AUSGABE 

9

AMIGA

Markt&Technik

DM 16,–
ÖS 120,– /Sfr 16,–
Lit 16000 /hfl 21,– /dkr 75,–

SONDERHEFT

Die besten LISTINGS des Jahres

20 Spitzenprogramme zum Sparpreis

Tolle Spiele ■ Faszinierende Grafikprogramme ■ Nützliche Anwendungen ■ Hilfreiche Tools

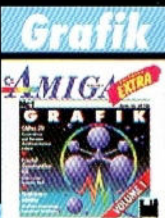
AMIGA Software EXTRA

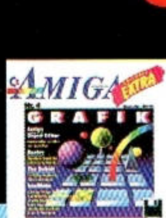
Markt&Technik

Software zum Taschengeldpreis

Jedes Programm nur DM 49,–* (sFr 45,–*/öS 490,–*)

Grafik





Amiga Extra Nr. 1: Grafik I
Drei Programme, die die außergewöhnlichen Grafikfähigkeiten des Amiga nutzen!
Bestell-Nr. 38708

Amiga Extra Nr. 4: Grafik II
Spielerisch Bobs erzeugen. IFF-Bilder werden zu Bobs und Images. Generieren von animierten Icons. Booter: par excellence.
Bestell-Nr. 38725

Spiele









Amiga Extra Nr. 12: Spiele
Highway 42: Als Kurier auf dem Planeten Cervezia. Warlords: Ein Brettspiel für taktisch geschickte Spieler. Zargon: Joystick-Action
Bestell-Nr. 38769

Amiga Extra Nr. 5: Spiele
Breaking out: Actionspiel mit toller Grafik und Sound. Decoder: Verwandeln Sie Ihren Amiga in eine Morsestation. Megamind:
Bestell-Nr. 38752

Amiga Extra Nr. 3: Spiele
Bliff: Eine ausgeklügelte Variante des Billards. Quadriga: Ein Spiel für Denker, angelehnt an »Vier gewinnt«. Wikinger I: ein Strategiespiel.
Bestell-Nr. 38724

Amiga Extra Nr. 13: Regnum
Regnum ist Ihr Königreich in einer imaginären Welt. Die Amiga-Maus dient Ihnen als Zepter. Ihre Aufgabe: Sichern und vergrößern Sie Ihren Besitz.
Bestell-Nr. 38781

Musik





Amiga Extra Nr. 6: Audio Worx
Ihr privates Sampling-Studio.
Bestell-Nr. 38748

Amiga Extra Nr. 9: Sonix-Hitkiste
Tolle, in Sonix editierbare und digitalisierte Geräusche und Effekte für eigene Musikstücke.
Bestell-Nr. 38753

Anwendungen und Utilities









Amiga Extra Nr. 2: Disk Utilities I
Disk-Ed V6, Select Copy, D Copy II, Check, Bootgirl Plus. Mit einem Super-Diskeditor.
Bestell-Nr. 38726

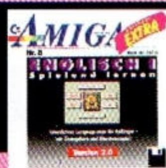
Amiga Extra Nr. 10: Disk-Utilities II
DIMO: Diskettenmonitor. Recover II: Datenrettung. TUC – The Ultimate Cruncher: Dateien extrem komprimieren.
Bestell-Nr. 38766

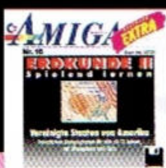
Amiga Extra Nr. 14: MenuMind 1.0
MenuMind bietet Ihnen eine menügesteuerte, grafische Programmierumgebung, die den CLI völlig ersetzt.
Bestell-Nr. 38771

Amiga Extra Nr. 11: Karteikasten
Finden Sie sich in Ihrem Diskettenbestand nicht mehr zurecht? Oder, oder, oder... Mit Dateiverwaltungen werden Ihre Probleme gelöst.
Bestell-Nr. 38768

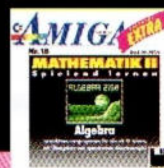
Spielend lernen!

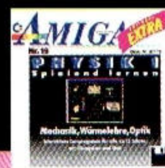












Amiga Extra Nr. 7: Erdkunde I
Die Reihe »Spielend lernen« verknüpft das Begeisternde des Amiga mit dem Nützlichen. Stupide Paukerei wird durch Kurzweil ersetzt.
Bestell-Nr. 38774

Amiga Extra Nr. 8: Englisch I
»Englisch I« vermittelt Ihnen Grundkenntnisse der englischen Sprache.
Bestell-Nr. 38775

Amiga Extra Nr. 16: Erdkunde II
Spielend lernen! Vereinigte Staaten von Amerika. Interaktives Lernprogramm für alle ab 12 Jahren, mit Übungskurs und Quiz.
Bestell-Nr. 38776

Amiga Extra Nr. 17: Mathematik I (Geometrie)
»Mathematik I« vermittelt Ihnen die Grundlagen der Geometrie bis hin zur sphärischen Trigonometrie.
Bestell-Nr. 38777

Amiga Extra Nr. 18: Mathematik II (Algebra)
vermittelt Ihnen die Grundlagen der Algebra, die die Voraussetzung für die gesamte weiterführende Mathematik sind.
Bestell-Nr. 38778

Amiga Extra Nr. 19: Physik I
Grundlagen der Mechanik, der Wärmelehre und der Optik. Animationen und Soundeffekte erhöhen die Verständlichkeit und die Lerneffizienz.
Bestell-Nr. 38779

INFO-COUPON

Bitte senden Sie mir Ihr Gesamtverzeichnis mit über 500 aktuellen Computerbüchern und Software

Name

Straße

PLZ/Ort

Bitte ausschneiden und schicken an: Markt&Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, 8013 Haar bei München

Ami SH 9

*Unverbindliche Preisempfehlung

Markt&Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computer-Fachgeschäften und in den Fachabteilungen der Warenhäuser

Markt&Technik
Zeitschriften · Bücher
Software · Schulung

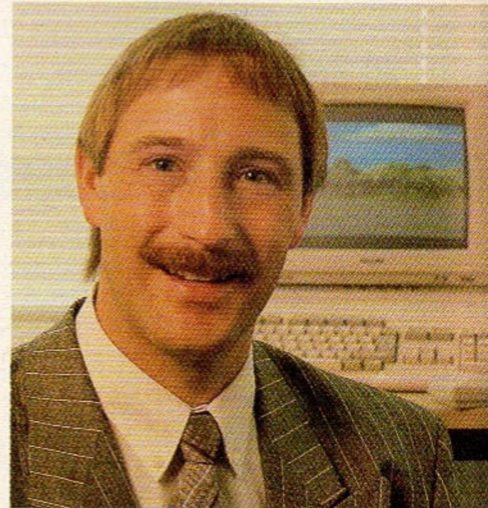
2210/912

Software kompakt

Ein prall gefülltes Software-Paket – und nur Spitzenprogramme. Wer wünscht sich das nicht?
Für Sie haben wir die besten Listings des Jahres 1989 zusammengestellt. Viele »Programme des Monats« vom AMIGA-Magazin und die »besten Stükke« aus den Sonderheften sind dabei.

Was hat uns dazu bewogen? Nun, zu Weihnachten bekamen Tausende einen neuen Amiga. Die Freude war sicher groß, doch besonders zu Anfang fehlt es an guter Software. Dem wollen wir abhelfen. Aus den verschiedensten Bereichen haben wir Spitzen-Listings zusammengestellt: faszinierende Grafik-Software, unterhaltsame Spiele, nützliche Anwendungen und hilfreiche Tools, die einem den Computeralltag erleichtern. Für jeden ist etwas dabei. Stöbern Sie nur in diesem Software-Paket, es lohnt sich.

Noch ein weiterer Punkt bewog uns zu dieser Software-Sammlung. Gingen Ihnen auch schon die folgenden Gedanken durch den Kopf? »Mensch, das Programm des Monats hat mich in dieser Ausgabe unwahrscheinlich interessiert. Aber dafür gleich die Programmservice-Diskette kaufen – nein.« Und zum Abtippen hatte man weder Lust noch Zeit. Also ließ man es bleiben. Für mehr als 20 dieser Programm-Renner gibt man gern das Geld für die Beschaffung der Programmservice-Diskette aus.



Warum dann noch die Listings im Heft? Wir müssen auch an diejenigen Leser denken, bei denen der Geldbeutel sehr eng geschnürt ist.

Auch diese Amiga-Fans sollen die Chance erhalten, in den Genuß der Software zu kommen – und meist steht Ihnen etwas mehr Zeit zur Verfügung.

Natürlich interessiert uns Ihre Meinung zu dieser Sammlung. Schreiben Sie uns, an die »Redaktion Sonderheft«. Wir warten mit Spannung auf Ihre Reaktionen.
Also, viel Spaß beim Fliegen mit dem Computer, Entwickeln von Riesen-Sprites, Tischtennis-Kampf gegen den elektronischen Gegner, Fiebern an der Börse und, und, und…

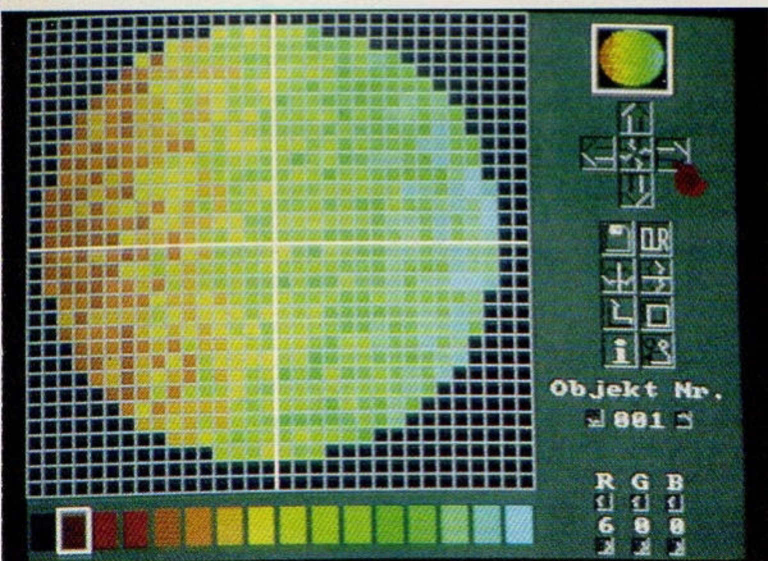
Ihr
Gottfried Knechtel
(Stellv. Chefredakteur)

Gottfried Knechtel

Grafik

Spiele





Mit dem mausgesteuerten »Amiga-Objekt-Editor« lassen sich Riesen-Sprites und Animationssequenzen schnell und mühelos erstellen. Ein Programm, das jeder Grafik-Fan braucht.

SEITE 6

Tools

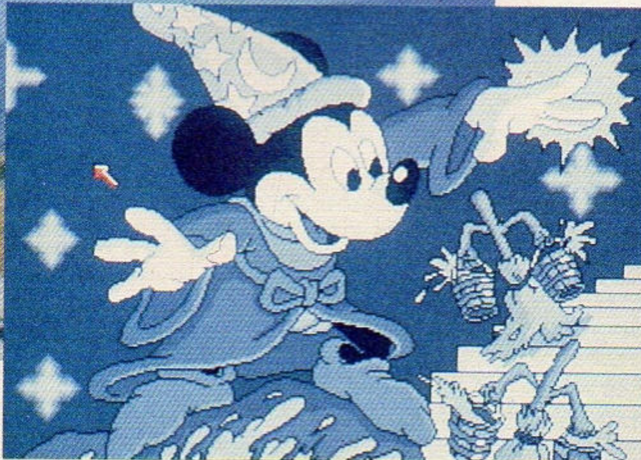


Wollten Sie schon immer hoch in der Luft schweben und Loopings drehen. Unser Flugsimulator macht es möglich – auch ohne Flugschein.

SEITE **40**

Sonstiges

Alle Programme aus Artikeln mit einem ■-Symbol finden Sie auch auf der Programmservice-Diskette zu diesem Sonderheft.



Solche Bilder können Sie mit dem »IFF-Filter« in den verschiedensten Varianten modifizieren. Wie – das erfahren Sie auf

SEITE **21**

Erweiterung

Anwendung



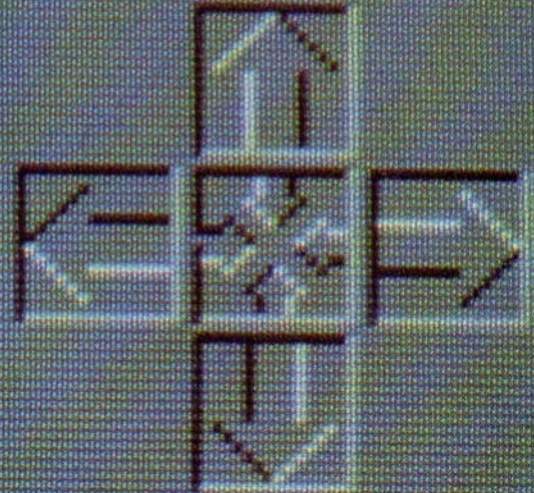
Jetzt kommt Ordnung in Ihre Karteien. Die Dateiverwaltung »Uni-Datei« ist derart benutzerfreundlich und schnell, daß keiner mehr darauf verzichten will.

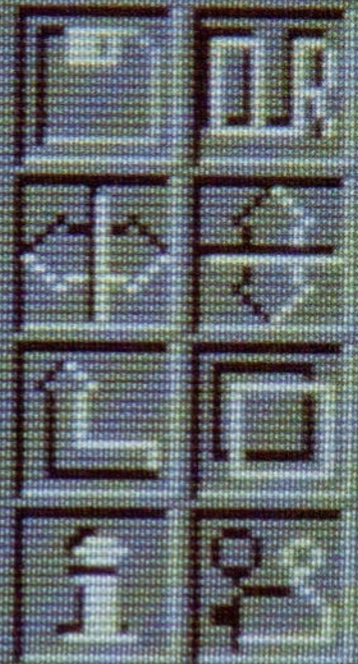
SEITE **140**

Sprite-Editor

Sprites gehören zu Spielen wie der Monitor zum Computer. Leider müssen diese auch auf dem Amiga in mühevoller Kleinarbeit erstellt werden. Unser Sprite-Editor beseitigt dieses Problem und macht selbst das Entwickeln von überdimensionalen Sprites zum Kinderspiel.







Objekt Nr

001

R G B

Es gibt kaum Spiele, die ohne Sprites auskommen. Dabei beschränkt sich die Verwendung von bewegten Objekten keineswegs nur auf Actionspiele. Auch Brettspiele, Knobelspiele, Adventures etc. profitieren ungemein durch den Einsatz beweglicher Objekte. Mit der mausgesteuerten Benutzeroberfläche des »Amiga-Object-Editor« (Bild 1) lassen sich Sprites und Animationssequenzen schnell und mühelos erstellen und editieren. Auch DPaint-Brushes lassen sich damit bearbeiten.

Geben Sie den DATA-Lader »AOE__Gen« (Listing 1) mit dem Checksummer ein. Da das Programm einen relativ hohen Speicherbedarf hat, muß vor der Eingabe des Listings im Direktmodus des Basic-Editors »CLEAR, 150000&« eingegeben werden. Dies hat auch dann zu geschehen, wenn »AOE__Gen« erneut in den Speicher geladen wird. Nach dem Start des Laders generiert dieser das lauffähige Programm »AOE__V1.2« auf Diskette.

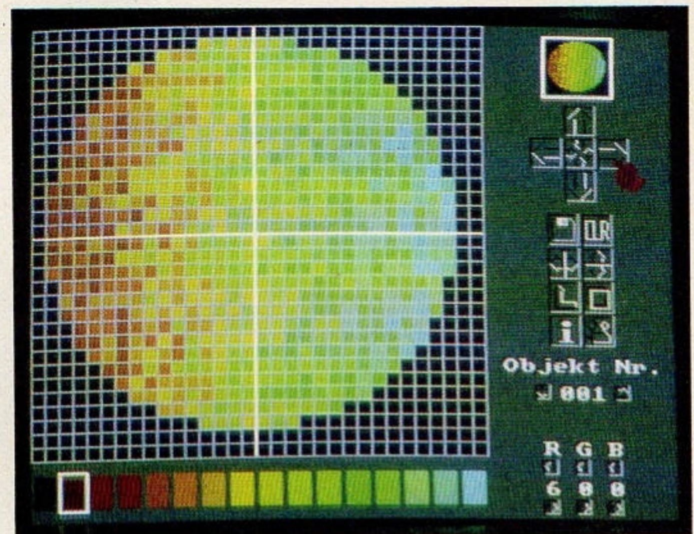


Bild 1. Die Benutzeroberfläche des »Amiga-Object-Editors« mit den verschiedenen Icons

Nach diesem Vorgang, der etwas Zeit beansprucht, kann der Object-Editor gestartet werden. Dabei ist zu beachten, daß das vom Lader generierte File nur vom CLI aus gestartet werden kann, obwohl ein Icon vorhanden wäre.

Sie werden nach dem Start aufgefordert, die Anzahl der Objekte, die bearbeitet werden sollen, einzugeben. Klicken Sie dazu das Eingabefeld an und tragen die gewünschte Objektanzahl ein.

Das Zeichnen eines Objektes ist durch die Maussteuerung sehr einfach. Es stehen 16 verschiedene Farben zur Verfügung, die durch einfaches Anklicken ausgewählt werden. Die Rot-Grün- und Blauanteile der gewählten Farbe können durch die Schalter am unteren rechten Bildschirmrand verändert werden. Gezeichnet wird ein Sprite mit Hilfe der Maus, wobei mit der linken Maustaste Punkte gesetzt und mit der rechten gelöscht werden. Das zu bearbeitende Sprite wird bei »Objekt Nr.« ausgewählt.

Im Zeichenfenster wird das Riesensprite, das in Wirklichkeit aus vier Einzelsprites besteht, stark vergrößert dargestellt. Rechts oben sieht man das Sprite in Originalgröße. Unter diesem Fenster befinden sich vier Pfeile. Durch Anklicken dieser Pfeile kann das aktuelle Sprite in alle vier Himmelsrichtungen verschoben

Icons erleichtern die Arbeit

werden. Klicken Sie das Icon in der Mitte der Pfeile an, so ändern diese ihre Richtung und das Sprite wird nun bei der Verwendung der Pfeilsymbole in Richtung seiner Mittelachse gestaucht. Unter den Pfeil-Icons befinden sich acht weitere Icons, auf deren Funktionen nun eingegangen werden soll:

Icon 1 (Disketten-Icon): Hier haben Sie die Möglichkeit, Sprites zu laden und zu speichern. Zum Laden muß die entsprechende Option gewählt und die Nummer, unter der das

Adresse	Wert	Bemerkung
$00000000	"AOE ".L	Kennung der Objektdatei
$00000004	0000.W	Anzahl der Objekte in der Datei
$00000006	0000.W	Nun folgen 16 Wörter mit der Farbinfomation
$00000026		Es folgen die Objektdaten: Anzahl der Objekte mal 640 Byte.
Die Objektdaten haben folgenden Aufbau:		
1. Objekt:		1. Bitplane (128 Byte)
		.....
		4. Bitplane (128 Byte)
		5. Maske (128 Byte)
2. Objekt:		1. Bitplane (128 Byte)
		.....
Jedem gesetzten Bit in der Maske entspricht ein gesetzter Punkt im Sprite		

Tabelle 1. Der Aufbau eines Files im »AOE«-Format

Objekt im Speicher stehen soll, sowie der Filename eingegeben werden.

Das Programm erkennt, ob es sich um ein »AOE«-Objekt (Amiga-Object-Editor-Objekt), oder um einen DPaint-Brush handelt. DPaint-Brushes müssen in der Größe von 32 x 32 Punkten mit 16 Farben vorliegen. Wollen Sie selbsterstellte Sprites auf Diskette ablegen, so wählen Sie »SPEICHERN«. Hier sind die Nummern des ersten und des letzten zu speichernden Objekts angegeben. Es können also beliebig viele aufeinanderfolgende Sprites in einem File gespeichert werden. Wollen Sie nur ein Sprite speichern, geben Sie einfach zwei gleiche Werte an. Nach Eingabe des Dateinamens werden die Objekte im »AOE«-Format gespeichert. Den Aufbau eines »AEO«-Files sehen Sie in Tabelle 1.

Kopieren und Animieren

Icon 2 (CLR): Wenn Sie dieses Icon anwählen, wird das aktuelle Sprite gelöscht.

Icon 3 (zwei waagerechte Pfeile): Horizontales Spiegeln des aktuellen Sprites.

Icon 4 (zwei senkrechte Pfeile): Vertikales Spiegeln des aktuellen Sprites.

Icon 5 (geknickter Pfeil): Drehen des aktuellen Sprites um 90 Grad.

Icon 6 (zwei sich überlappende Quadrate): Kopieren von Sprites. Neben der Quell- und Zielnummer des Objekts kann beim Kopieren von mehreren aufeinanderfolgenden Sprites auch die gewünschte Anzahl eingegeben werden.

Icon 7 (»i«): Info zum Programm.

Icon 8 (Bildprojektor-Symbol): Animation von mehreren Sprites. Geben Sie die Nummern des ersten und letzten Sprites der Animationsfolge ein und klicken Sie »FERTIG« an. Daraufhin beginnt die Animation, deren Geschwindigkeit mit der Maus verändert werden kann. Die Zahl zwischen den Pfeilen gibt die Geschwindigkeit in 1/50-Sekunden an.

Haben Sie Ihre Arbeit beendet und wollen den Editor verlassen, bewegen Sie die Maus in die rechte obere Bildschirmecke und drücken den rechten Mausknopf.

Nun steht dem munteren Erstellen von Sprites und Animationssequenzen nichts mehr im Wege. Grenzen für neue Objekte setzt nur die eigene Kreativität — und davon haben Amiga-Besitzer, wie wir wissen, eine ganze Menge.

Michael Bertsch/C. Ewald/ M. Jobst/kn

Programmname:	AOE_Gen
Computer:	Amiga 500, 2000 mit Kickstart 1.2, 1.3
Sprache:	Amiga-Basic 1.2

So geben Sie das Listing ein

Der Objekt-Editor ist in Maschinensprache (Assembler) geschrieben. Um jedem Amiga-Besitzer die Eingabe des Listings zu ermöglichen, haben wir das Programm für Sie umgewandelt. Die abgedruckte Version ist mit Amiga-Basic einzugeben. Diesen Interpreter finden Sie auf der Extras-Diskette, die bei der Auslieferung jedem Amiga beiliegt. Das Programm erzeugt beim Start das lauffähige Programm mit Namen »AOE__V1.2« auf der eingelegten Diskette. Bitte geben Sie das Programm unbedingt mit Checkie 42 (Seite 143) ein. Die Hinweise zum Checksummer sollten Sie unbedingt beachten.

Listing 1. Der Quellcode des Object-Editors als DATA-Lader. Mit den Objekt-Editor erzeugen Sie Sprites und Animationssequenzen. Bitte mit dem Checksummer auf Seite 143 eingeben.

```
tI0 REM Generiert Programm AOE_V1.2
ag  CLS
g0  OPEN "AOE_V1.2" FOR OUTPUT AS 1
BS  READ anz
oa  FOR i=1 TO anz
3n1  READ h$
yB2   wert1=ASC(LEFT$(h$,1))
FD    IF wert1>64 THEN wert1=wert1-55 ELSE wert1=wert1-48
FI    wert1=wert1*16
7c    wert2=ASC(RIGHT$(h$,1))
ad    IF wert2>64 THEN wert2=wert2-55 ELSE wert2=wert2-48
Pi    wert=wert1+wert2
9G    PRINT #1,CHR$(wert);
JOO NEXT
3n  CLOSE 1
0v  END
R4  DATA 19072
ie  DATA 00,00,03,F3,00,00,00,00,00,00
aj  DATA 00,02,00,00,00,00,00,00,00,01
UF  DATA 00,00,0F,1A,00,00,01,00,00,00
gb  DATA 03,E9,00,00,0F,1A,4E,F9,00,00
72  DATA 2B,94,2C,78,00,04,43,F9,00,00
ut  DATA 3B,30,42,80,4E,AE,FD,D8,23,C0
Sy  DATA 00,00,3A,3E,43,F9,00,00,3B,41
az  DATA 42,80,4E,AE,FD,D8,23,C0,00,00
6d  DATA 3A,42,43,F9,00,00,3B,4D,42,80
6p  DATA 4E,AE,FD,D8,23,C0,00,00,3A,46
N4  DATA 4E,75,2C,78,00,04,22,79,00,00
D7  DATA 3A,3E,4E,AE,FE,62,22,79,00,00
Tc  DATA 3A,42,4E,AE,FE,62,22,79,00,00
bf  DATA 3A,46,4E,AE,FE,62,4E,75,2C,79
KH  DATA 00,00,3A,46,4E,AE,FF,B2,41,F9
gd  DATA 00,00,2D,46,4E,AE,FF,3A,23,C0
9F  DATA 00,00,2D,84,23,C0,00,00,2D,B4
zt  DATA 23,C0,00,00,2D,E4,41,F9,00,00
PL  DATA 2D,66,4E,AE,FF,34,23,C0,00,00
fb  DATA 3A,52,20,40,4E,AE,FE,D4,23,C0
cK  DATA 00,00,3A,62,20,79,00,00,3A,52
W3  DATA 23,E8,00,32,00,00,3A,66,23,E8
BS  DATA 00,32,00,00,3A,6E,20,68,00,32
ge  DATA 23,E8,00,04,00,00,3A,5A,2C,79
4I  DATA 00,00,3A,3E,20,79,00,00,3A,62
v1  DATA 43,F9,00,00,32,6E,30,3C,00,20
2u  DATA 4E,AE,FF,40,2C,79,00,00,3A,46
oE  DATA 4E,AE,FE,80,4E,75,2C,79,00,00
kU  DATA 3A,46,20,79,00,00,3A,52,4E,AE
QW  DATA FF,B8,20,79,00,00,2D,84,4E,AE
63  DATA FF,BE,4E,AE,FF,2E,4E,75,2C,78
2p  DATA 00,04,93,C9,4E,AE,FE,DA,28,40
yZ  DATA 4A,AC,00,AC,66,00,00,18,41,EC
pR  DATA 00,5C,4E,AE,FE,80,41,EC,00,5C
h0  DATA 4E,AE,FE,8C,23,C0,00,00,3A,82
aC  DATA 4E,75,4A,B9,00,00,3A,82,67,00
A1  DATA 00,0C,22,79,00,00,3A,82,4E,AE
HD  DATA FE,86,42,80,4E,75,2C,79,00,00
bd  DATA 3A,3E,43,F9,00,00,3A,98,4E,AE
71  DATA FF,3A,23,FC,00,00,3B,08,00,00
C1  DATA 3A,9C,22,79,00,00,3A,66,30,3C
fz  DATA 00,10,4E,AE,FE,AA,22,79,00,00
8j  DATA 3A,66,42,80,42,81,34,3C,00,E0
WD  DATA 36,3C,00,E0,4E,AE,FE,CE,22,79
Dd  DATA 00,00,3A,66,30,3C,00,02,4E,AE
UQ  DATA FE,AA,22,79,00,00,3A,66,42,80
RH  DATA 4E,AE,FE,A4,22,79,00,00,3A,66
jK  DATA 10,3C,00,00,4E,AE,FE,9E,42,86
H1  DATA 22,79,00,00,3A,66,33,46,00,24
nV  DATA 33,7C,00,00,00,26,30,06,32,3C
xL  DATA 00,E0,4E,AE,FF,0A,22,79,00,00
DZ  DATA 3A,66,33,7C,00,00,00,24,33,46
HO  DATA 00,26,30,3C,00,E0,32,06,4E,AE
AZ  DATA FF,0A,DC,7C,00,07,BC,7C,00,E7
9q  DATA 66,C2,22,79,00,00,3A,66,30,3C
CF  DATA 00,04,4E,AE,FE,AA,22,79,00,00
1U  DATA 3A,66,33,7C,00,70,00,24,33,7C
KC  DATA 00,01,00,26,30,3C,00,70,32,3C
85  DATA 00,DF,4E,AE,FF,0A,22,79,00,00
dG  DATA 3A,66,33,7C,00,01,00,24,33,7C
tV  DATA 00,70,00,26,30,3C,00,DF,32,3C
0A  DATA 00,70,4E,AE,FF,0A,22,79,00,00
BR  DATA 3A,66,30,3C,00,01,4E,AE,FE,AA
08  DATA 22,79,00,00,3A,66,30,3C,00,E1
E1  DATA 42,81,34,3C,01,3F,36,3C,00,FF
UT  DATA 4E,AE,FE,CE,22,79,00,00,3A,66
8H  DATA 30,3C,00,00,32,3C,00,E1,34,3C
d1  DATA 00,E1,36,3C,00,FF,4E,AE,FE,CE
oc  DATA 22,79,00,00,3A,66,30,3C,00,02
EH  DATA 4E,AE,FE,AA,22,79,00,00,3A,66
3i  DATA 30,3C,00,01,4E,AE,FE,A4,30,3C
nL  DATA 00,02,38,3C,00,FF,3A,3C,00,07
H1  DATA 3C,3C,01,20,3E,3C,00,28,4E,B9
VC  DATA 00,00,04,88,30,3C,00,04,38,3C
QS  DATA 00,FE,3A,3C,00,06,3C,3C,01,21
R5  DATA 3E,3C,00,29,4E,B9,00,00,04,88
qd  DATA 22,79,00,00,3A,66,30,3C,00,10
MP  DATA 4E,AE,FE,AA,22,79,00,00,3A,66
p1  DATA 30,3C,01,00,32,3C,00,08,34,3C
v5  DATA 01,1F,36,3C,00,27,4E,AE,FE,CE
fw  DATA 4B,F9,00,00,32,AE,3C,3C,00,10
tP  DATA 30,06,22,79,00,00,3A,66,4E,AE
jx  DATA FE,AA,30,1D,32,1D,34,1D,36,1D
Xn  DATA 22,79,00,00,3A,66,4E,AE,FE,CE
II  DATA DC,7C,00,01,BC,7C,00,20,66,D8
kE  DATA 30,3C,00,02,38,3C,00,0F,3A,3C
1U  DATA 00,E7,3C,3C,00,1B,3E,3C,00,F9
4E  DATA 4E,B9,00,00,04,88,30,3C,00,04
Zk  DATA 38,3C,00,0E,3A,3C,00,E6,3C,3C
0i  DATA 00,1C,3E,3C,00,FA,4E,B9,00,00
Uh  DATA 04,88,2C,79,00,00,3A,46,20,79
WR  DATA 00,00,3A,66,43,F9,00,00,34,36
mH  DATA 30,3C,01,09,32,3C,00,50,4E,AE
Tf  DATA FF,8E,43,F9,00,00,34,FE,20,79
D2  DATA 00,00,3A,66,30,3C,01,00,32,3C
Ue  DATA 00,66,4E,AE,FF,8E,43,F9,00,00
an  DATA 35,9E,20,79,00,00,3A,66,30,3C
fH  DATA 00,F8,32,3C,00,BC,4E,AE,FF,8E
NL  DATA 43,F9,00,00,35,C6,20,79,00,00
pZ  DATA 3A,66,30,3C,00,FB,32,3C,00,E2
Di  DATA 4E,AE,FF,8E,4B,F9,00,00,3B,5F
Rn  DATA 3A,3C,00,0A,3C,3C,00,E8,3E,3C
QW  DATA 00,B4,4E,B9,00,00,04,18,4B,F9
Ts  DATA 00,00,3B,74,3A,3C,00,03,3C,3C
EI  DATA 01,03,3E,3C,00,C2,4E,B9,00,00
Eh  DATA 04,18,4B,F9,00,00,3B,69,3A,3C
yj  DATA 00,05,3C,3C,00,FB,3E,3C,00,DE
2I  DATA 4E,B9,00,00,04,18,4E,B9,00,00
```

```
R3 DATA 05,C2,4E,75,2C,79,00,00,3A,3E
zi DATA 22,79,00,00,3A,6E,33,46,00,24
ps DATA 33,47,00,26,30,3C,00,01,4E,AE
G3 DATA FE,9E,22,79,00,00,3A,6E,42,80
8Q DATA 4E,AE,FE,AA,22,79,00,00,3A,6E
mS DATA 20,4D,30,05,4E,AE,FF,C4,22,79
GK DATA 00,00,3A,6E,DC,7C,00,01,DE,7C
Db DATA 00,01,33,46,00,24,33,47,00,26
b4 DATA 42,80,4E,AE,FE,9E,22,79,00,00
cw DATA 3A,6E,30,3C,00,03,4E,AE,FE,AA
up DATA 22,79,00,00,3A,6E,20,4D,30,05
YL DATA 4E,AE,FF,C4,4E,75,2C,79,00,00
07 DATA 3A,3E,22,79,00,00,3A,6E,4E,AE
gw DATA FE,AA,22,79,00,00,3A,6E,33,44
Ra DATA 00,24,33,45,00,26,30,06,32,05
j8 DATA 4E,AE,FF,0A,22,79,00,00,3A,6E
Qv DATA 30,06,32,07,4E,AE,FF,0A,22,79
wJ DATA 00,00,3A,6E,30,04,32,07,4E,AE
5y DATA FF,0A,22,79,00,00,3A,6E,30,04
S1 DATA 32,05,4E,AE,FF,0A,4E,75,42,87
yx DATA 41,F9,00,00,32,AE,30,18,32,18
3p DATA 34,18,36,18,B0,45,62,00,00,1A
13 DATA B2,46,62,00,00,14,B4,45,65,00
no DATA 00,0E,B6,46,65,00,00,08,4E,F9
XR DATA 00,00,05,20,DE,7C,00,01,BE,7C
ds DATA 00,10,66,D0,08,39,00,06,00,BF
HY DATA E0,01,67,F6,60,00,26,FE,23,F9
C9 DATA 00,00,3A,66,00,00,3A,6E,33,C7
PP DATA 00,00,3A,74,3E,39,00,00,3A,72
1v DATA CE,FC,00,08,28,47,D9,FC,00,00
N8 DATA 32,AE,61,00,00,62,61,00,00,68
00 DATA 30,3C,00,01,61,00,FF,38,61,00
yZ DATA 00,5C,30,3C,00,01,61,00,FF,2C
rd DATA 42,87,3E,39,00,00,3A,74,33,C7
ug DATA 00,00,3A,72,CE,FC,00,08,28,47
SN DATA D9,FC,00,00,32,AE,61,00,00,2C
AR DATA 61,00,00,32,30,3C,00,02,61,00
Fn DATA FF,02,61,00,00,26,30,3C,00,04
b7 DATA 61,00,FE,F6,61,00,00,2C,08,39
IQ DATA 00,06,00,BF,E0,01,67,F6,60,00
Bq DATA 26,78,38,1C,3A,1C,3C,1C,3E,1C
CH DATA 4E,75,98,7C,00,01,9A,7C,00,01
pz DATA DC,7C,00,01,DE,7C,00,01,4E,75
TU DATA 42,80,30,39,00,00,3A,72,C0,FC
HY DATA 00,02,22,40,D3,FC,00,00,32,8E
8N DATA 30,11,3E,00,61,00,00,48,13,C0
98 DATA 00,00,3B,72,30,07,E8,48,61,00
wu DATA 00,3A,13,C0,00,00,3B,70,30,07
Dw DATA E0,48,61,00,00,2C,13,C0,00,00
HS DATA 3B,6E,4B,F9,00,00,3B,6E,3A,3C
Vv DATA 00,06,3C,3C,00,FB,3E,3C,00,F2
gp DATA 23,F9,00,00,3A,66,00,00,3A,6E
vb DATA 4E,B9,00,00,04,18,4E,75,C0,7C
x3 DATA 00,0F,D0,7C,00,30,B0,7C,00,3A
L0 DATA 65,00,00,06,D0,7C,00,07,4E,75
Ao DATA 2C,79,00,00,3A,3E,8A,FC,00,07
5q DATA 8C,FC,00,07,CA,BC,00,00,FF,FF
yR DATA CC,BC,00,00,FF,FF,20,05,22,06
or DATA C0,FC,00,07,C2,FC,00,07,D0,7C
iq DATA 00,01,D2,7C,00,01,34,00,36,01
D4 DATA D4,7C,00,05,D6,7C,00,05,48,E7
jZ DATA F0,00,30,39,00,00,3A,72,D0,7C
Jb DATA 00,10,22,79,00,00,3A,66,4E,AE
cN DATA FE,AA,22,79,00,00,3A,66,4C,DF
4G DATA 00,0F,4E,AE,FE,CE,43,F9,00,00
Vw DATA 3A,98,30,39,00,00,3A,72,D0,7C
nK DATA 00,10,4E,AE,FE,AA,43,F9,00,00
Ws DATA 3A,98,30,05,32,06,4E,AE,FE,BC
Bc DATA 22,79,00,00,3A,66,30,05,32,06
EJ DATA D0,7C,01,00,D2,7C,00,08,4E,AE
eX DATA FE,BC,4E,75,42,87,41,F9,00,00
Hd DATA 33,2E,30,18,32,18,34,18,36,18
jP DATA B0,45,62,00,00,1A,B2,46,62,00
Bz DATA 00,14,B4,45,65,00,00,0E,B6,46
gY DATA 65,00,00,08,4E,F9,00,00,07,1A
qr DATA DE,7C,00,01,BE,7C,00,16,66,D0
4C DATA 08,39,00,06,00,BF,E0,01,67,F6
qn DATA 60,00,25,04,CE,FC,00,04,20,47
jc DATA D1,FC,00,00,33,DE,22,50,4E,D1
7F DATA 08,39,00,06,00,BF,E0,01,67,F6
9I DATA 23,F9,00,00,3A,66,00,00,3A,6E
hk DATA 60,00,24,DC,0C,79,00,01,00,00
2D DATA 3A,7C,67,00,01,84,20,79,00,00
DM DATA 3B,10,61,00,00,26,20,79,00,00
DI DATA 3B,14,61,00,00,1C,20,79,00,00
nL DATA 3B,18,61,00,00,12,20,79,00,00
eY DATA 3B,1C,61,00,00,08,61,00,20,2E
Vd DATA 60,AE,2E,28,00,7C,24,48,22,48
7v DATA D3,FC,00,00,00,80,D1,FC,00,00
hI DATA 00,7C,20,3C,00,00,00,1E,23,20
5d DATA 51,C8,FF,FC,24,87,4E,75,0C,79
bR DATA 00,01,00,00,3A,7C,67,00,01,78
3Q DATA 20,79,00,00,3B,10,61,00,00,28
Fs DATA 20,79,00,00,3B,14,61,00,00,1E
5V DATA 20,79,00,00,3B,18,61,00,00,14
cP DATA 20,79,00,00,3B,1C,61,00,00,0A
hU DATA 61,00,1F,D0,60,00,FF,50,2E,10
93 DATA 24,48,22,48,D1,FC,00,00,00,04
2z DATA 20,3C,00,00,00,1E,22,D8,51,C8
0x DATA FF,FC,25,47,00,7C,4E,75,0C,79
j1 DATA 00,01,00,00,3A,7C,67,00,01,6E
CZ DATA 20,79,00,00,3B,10,61,00,00,28
01 DATA 20,79,00,00,3B,14,61,00,00,1E
Ee DATA 20,79,00,00,3B,18,61,00,00,14
1Y DATA 20,79,00,00,3B,1C,61,00,00,0A
s2 DATA 61,00,1F,76,60,00,FE,F6,20,3C
Hb DATA 00,00,00,1F,22,10,E2,99,20,C1
Pc DATA 51,C8,FF,F8,4E,75,0C,79,00,01
9z DATA 00,00,3A,7C,67,00,01,62,20,79
ea DATA 00,00,3B,10,61,00,00,28,20,79
am DATA 00,00,3B,14,61,00,00,1E,20,79
dc DATA 00,00,3B,18,61,00,00,14,20,79
2t DATA 00,00,3B,1C,61,00,00,0A,61,00
1U DATA 1F,28,60,00,FE,A8,20,3C,00,00
GH DATA 00,1F,22,10,E3,99,20,C1,51,C8
rH DATA FF,F8,4E,75,0C,79,00,01,00,00
x4 DATA 3A,7C,67,00,01,5A,2C,79,00,00
0Q DATA 3A,46,20,79,00,00,3A,66,43,F9
fH DATA 00,00,34,9A,20,3C,00,00,01,09
wt DATA 22,3C,00,00,00,50,4E,AE,FF,8E
tN DATA 33,FC,00,01,00,00,3A,7C,60,00
fJ DATA FE,5C,20,79,00,00,3B,10,61,00
u0 DATA 00,28,20,79,00,00,3B,14,61,00
Mf DATA 00,1E,20,79,00,00,3B,18,61,00
R1 DATA 00,14,20,79,00,00,3B,1C,61,00
En DATA 00,0A,61,00,1E,AC,60,00,FE,2C
ie DATA 24,48,22,48,D3,FC,00,00,00,40
FU DATA D1,FC,00,00,00,44,20,3C,00,00
AM DATA 00,0F,22,D8,51,C8,FF,FC,42,AA
dD DATA 00,7C,4E,75,20,79,00,00,3B,10
si DATA 61,00,00,28,20,79,00,00,3B,14
Xi DATA 61,00,00,1E,20,79,00,00,3B,18
ty DATA 61,00,00,14,20,79,00,00,3B,1C
Tr DATA 61,00,00,0A,61,00,1E,5A,60,00
8J DATA FD,DA,24,48,22,48,D3,FC,00,00
Bm DATA 00,3C,D1,FC,00,00,00,40,20,3C
zk DATA 00,00,00,0F,21,21,51,C8,FF,FC
VJ DATA 42,92,4E,75,20,79,00,00,3B,10
Oq DATA 61,00,00,28,20,79,00,00,3B,14
fq DATA 61,00,00,1E,20,79,00,00,3B,18
16 DATA 61,00,00,14,20,79,00,00,3B,1C
Mf DATA 61,00,00,0A,61,00,1E,0A,60,00
5U DATA FD,8A,20,3C,00,00,00,1F,22,10
8J DATA E3,49,20,C1,51,C8,FF,F8,4E,75
tG DATA 20,79,00,00,3B,10,61,00,00,28
5i DATA 20,79,00,00,3B,14,61,00,00,1E
vL DATA 20,79,00,00,3B,18,61,00,00,14
SF DATA 20,79,00,00,3B,1C,61,00,00,0A
p1 DATA 61,00,1D,C8,60,00,FD,48,20,3C
09 DATA 00,00,00,1F,22,10,48,41,E2,49
uL DATA 48,41,20,C1,51,C8,FF,F4,4E,75
xr DATA 2C,79,00,00,3A,46,20,79,00,00
Me DATA 3A,66,43,F9,00,00,34,36,20,3C
UP DATA 00,00,01,09,22,3C,00,00,00,50
4w DATA 4E,AE,FF,8E,33,FC,00,00,00,00
RF DATA 3A,7C,60,00,FD,04,41,F9,00,00
Vi DATA 2D,96,2C,79,00,00,3A,46,4E,AE
Qt DATA FF,34,23,C0,00,00,3A,56,67,00
kJ DATA FC,EA,20,40,23,E8,00,32,00,00
bU DATA 3A,6A,22,68,00,32,30,3C,00,01
W9 DATA 2C,79,00,00,3A,3E,4E,AE,FE,AA
VT DATA 22,79,00,00,3A,6A,30,3C,00,01
wx DATA 4E,AE,FE,A4,22,79,00,00,3A,6A
M9 DATA 30,3C,00,00,32,3C,00,00,34,3C
po DATA 00,DF,36,3C,00,DF,4E,AE,FE,CE
3N DATA 23,F9,00,00,3A,6A,00,00,3A,6E
EZ DATA 4B,F9,00,00,3B,77,3A,3C,00,05
BE DATA 3C,3C,00,5C,3E,3C,00,5B,4E,B9
4p DATA 00,00,04,18,4B,F9,00,00,3B,7C
u2 DATA 3A,3C,00,09,3C,3C,00,4C,3E,3C
qi DATA 00,6D,4E,B9,00,00,04,18,4B,F9
1s DATA 00,00,3C,13,3A,3C,00,07,3C,3C
6I DATA 00,54,3E,3C,00,7F,4E,B9,00,00
9D DATA 04,18,30,3C,00,02,38,3C,00,46
IO DATA 3A,3C,00,50,3C,3C,00,9A,3E,3C
E8 DATA 00,60,4E,B9,00,00,04,88,30,3C
rs DATA 00,02,38,3C,00,46,3A,3C,00,62
jG DATA 3C,3C,00,9A,3E,3C,00,72,4E,B9
oT DATA 00,00,04,88,30,3C,00,02,38,3C
nw DATA 00,46,3A,3C,00,74,3C,3C,00,9A
IT DATA 3E,3C,00,84,4E,B9,00,00,04,88
lt DATA 08,39,00,06,00,BF,E0,01,67,F6
kr DATA 08,39,00,06,00,BF,E0,01,66,F6
rg DATA 22,79,00,00,3A,56,3A,29,00,0E
fs DATA 3C,29,00,0C,BA,7C,00,46,65,00
CX DATA 00,1E,BC,7C,00,50,65,00,00,16
Vw DATA BA,7C,00,9A,62,00,00,0E,BC,7C
DO DATA 00,60,62,00,00,06,60,00,00,52
zd DATA BA,7C,00,46,65,00,00,1E,BC,7C
Jr DATA 00,62,65,00,00,16,BA,7C,00,9A
1Y DATA 62,00,00,0E,BC,7C,00,72,62,00
cq DATA 00,06,60,00,04,F4,BA,7C,00,46
gJ DATA 65,9A,BC,7C,00,74,65,94,BA,7C
g3 DATA 00,9A,62,8E,BC,7C,00,84,62,88
eY DATA 2C,79,00,00,3A,46,20,79,00,00
lt DATA 3A,56,4E,AE,FF,B8,60,00,FB,70
ga DATA 2C,79,00,00,3A,46,20,79,00,00
EZ DATA 3A,56,4E,AE,FF,B8,20,79,00,00
O4 DATA 3A,62,30,3C,00,1F,32,3C,00,0D
Fm DATA 34,01,36,01,2C,79,00,00,3A,3E
Rr DATA 4E,AE,FE,E0,20,79,00,00,3A,62
I4 DATA 30,3C,00,1E,32,3C,00,03,34,01
MR DATA 36,01,4E,AE,FE,E0,2C,79,00,00
i8 DATA 3A,46,23,FC,00,00,2D,F6,00,00
xo DATA 2D,A8,41,F9,00,00,2D,96,4E,AE
qW DATA FF,34,23,C0,00,00,3A,56,20,40
xf DATA 23,E8,00,32,00,00,3A,6A,23,E8
6F DATA 00,32,00,00,3A,6E,22,68,00,32
er DATA 30,3C,00,01,2C,79,00,00,3A,3E
7L DATA 4E,AE,FE,AA,22,79,00,00,3A,6A
T8 DATA 30,3C,00,01,4E,AE,FE,A4,22,79
6A DATA 00,00,3A,6A,30,3C,00,00,32,3C
Jd DATA 00,00,34,3C,00,DF,36,3C,00,54
Vf DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
kh DATA 30,3C,00,00,32,3C,00,55,34,3C
m5 DATA 00,5D,36,3C,00,61,4E,AE,FE,CE
u0 DATA 22,79,00,00,3A,6A,30,3C,00,81
au DATA 32,3C,00,55,34,3C,00,DF,36,3C
1S DATA 00,61,4E,AE,FE,CE,22,79,00,00
w4 DATA 3A,6A,30,3C,00,00,32,3C,00,61
cD DATA 34,3C,00,DF,36,3C,00,72,4E,AE
NW DATA FE,CE,22,79,00,00,3A,6A,30,3C
cT DATA 00,00,32,3C,00,73,34,3C,00,24
WT DATA 36,3C,00,7F,4E,AE,FE,CE,22,79
hB DATA 00,00,3A,6A,30,3C,00,B8,32,3C
xW DATA 00,73,34,3C,00,DF,36,3C,00,7F
is DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
9P DATA 30,3C,00,00,32,3C,00,7F,34,3C
rq DATA 00,DF,36,3C,00,DF,4E,AE,FE,CE
vC DATA 4B,F9,00,00,3B,85,3A,3C,00,11
jp DATA 3C,3C,00,27,3E,3C,00,4F,4E,B9
Ak DATA 00,00,04,18,4B,F9,00,00,3B,96
3x DATA 3A,3C,00,09,3C,3C,00,27,3E,3C
ff DATA 00,6B,4E,B9,00,00,04,18,4B,F9
Iu DATA 00,00,3B,9F,3A,3C,00,06,3C,3C
YB DATA 00,58,3E,3C,00,91,4E,B9,00,00
47 DATA 04,18,30,3C,00,02,38,3C,00,54
uY DATA 3A,3C,00,87,3C,3C,00,8C,3E,3C
Gf DATA 00,96,4E,B9,00,00,04,88,30,39
39 DATA 00,00,2E,82,08,00,00,07,66,00
Wv DATA 00,36,08,39,00,06,00,BF,E0,01
Hh DATA 66,E8,22,79,00,00,3A,56,3A,29
30 DATA 00,0E,3C,29,00,0C,BA,7C,00,54
t2 DATA 65,D4,BC,7C,00,87,65,CE,BA,7C
f6 DATA 00,8C,62,C8,BC,7C,00,96,62,C2
8R DATA 60,00,00,0E,30,39,00,00,2E,82
g0 DATA 08,00,00,07,66,F4,0C,B9,00,00
jV DATA 00,00,00,00,2E,62,66,00,00,0C
```

Listing 1. (Fortsetzung)

Superbase™ FÜR JEDEN GELDBEUTEL

ca. DM 89,–*
(sFr 81,90*/öS 757,–*)

BOOKWARE
Amiga Superbase

Amiga Superbase (Bookware)
Wegen seiner Bedienerfreundlichkeit die optimale Einsteiger-Datenbank. Per Mausklick erstellen Sie Ihre Datenmaske. Aufnahme, Auswahl und Ausgabe der Daten erfolgen über ein Bedienerfeld, das dem eines Videorecorders sehr ähnlich ist. Selbstverständlich können Sie Amiga-Grafiken ebenso verwalten und anzeigen lassen wie Zahlen und Texte – Briefmarkensammler zum Beispiel könnten somit die digitalisierten Abbilder ihrer Schätze mit den zugehörigen Daten auf den Monitor bringen. Das System ist relational; Sie können also Einträge verschiedener Datenbanken miteinander verknüpfen. Die Daten lassen sich als Formular oder Liste auf dem Bildschirm oder Drucker ausgeben. Vorbei ist also die Zeit des Chaos in Platten-, Dia- oder anderen Sortimenten!
Hardware-Anforderungen: Amiga 500, 1000, 2000 mit mindestens 512 Kbyte Arbeitsspeicher.
1989, 188 Seiten, inkl. Programmdiskette,
ISBN 3-89090-791-1, Bestell-Nr. 90791

DM 199,–*
(sFr 179,–*/öS 1990,–*)

AMIGA Superbase 2

Superbase 2 Amiga (deutsch)
Wegen seiner Verkaufszahlen der Renner unter den Datenbanken, für den Hausgebrauch ebenso geeignet wie für das Büro. Daten und Bilder lassen sich mit diesem mächtigen, relationalen Datenbanksystem fast spielerisch verwalten. Eine neue Form der Lagerhaltung ist somit beispielsweise möglich: Nicht mehr nur die Daten, sondern auch ein digitalisiertes Bild jedes Artikels erscheinen auf dem Monitor. Bis zu 16 Millionen Datensätze pro Datei und eine unbegrenzte Anzahl geöffneter Dateien sind erlaubt. Die Ein- oder Ausgabe erfolgt in Listen oder Formularen, die Sie sich am Bildschirm erstellen können. Weitere Leistungsmerkmale: Textverarbeitungsprogramm integriert, Serienbrieffunktion, speicherbare Masken, Etikettendruck u.v.m.
Hardware-Anforderungen: Amiga 500, 1000, 2000 mit mind. 512 Kbyte RAM (empfohlen 1 Mbyte).
Bestell-Nr. 54110
Upgrade von Superbase 2 auf Superbase Professional, 51672U, DM 199,–* (sFr 179,–*/öS 1990,–*)

DM 399,–*
(sFr 359,–*/öS 3690,–*)

AMIGA Superbase PROFESSIONAL

Superbase Professional
Neben den Funktionen von Superbase 2 bietet Ihnen diese Professional-Version zwei erhebliche Vorteile: zum einen den Formular-Editor, mit dem Sie sich grafische Ein- und Ausgabemasken in grandioser Qualität schaffen können, und zum anderen beinhaltet sie eine Programmiermöglichkeit. Die Programmiersprache DML ist zwar einfach erlernbar wie Basic, unterstützt die Möglichkeiten von Superbase jedoch vollständig. Somit sind der Flexibilität der Datenbank keine Grenzen gesetzt, sei es bei der individuellen Bedienung, sei es bei der Möglichkeit, Daten zu selektieren, darzustellen und auf dem Drucker auszugeben.
Hardware-Anforderungen: Amiga 500, 1000, 2000 mit mindestens 512 Kbyte RAM (empfohlen: 1 Mbyte RAM, Festplatte), Bestell-Nr. 51672
Upgrade Superbase Professional auf Superbase Professional Entwickler-Paket, Bestell-Nr. 54136U, gegen Einsendung der Originaldiskette DM 199,–* (sFr 179,–*/öS 1990,–*)

DM 599,–*
(sFr 539,–*/öS 5690,–*)

AMIGA Superbase PROFESSIONAL ENTWICKLER-PAKET

Superbase Professional Entwickler-Paket
Superbase Professional Entwickler-Paket ist die konsequente Weiterentwicklung der herausragenden Datenbank Superbase Professional. Im Anwenderteil wurden die Tastaturbedienung, die Einbaumöglichkeit von Hilfefenstern, die Import-/Export-Funktion, der Formulareditor u.v.m. erheblich erweitert. dBase, Lotus u.a. Dateien können nun komfortabel eingelesen werden. Ein leistungsstarkes DFÜ-Programm wurde integriert, um die Datenfernübertragung zu erleichtern. Für die Programmierer stehen nun weitere Befehle zur Verfügung, die zum Beispiel Transaktionen von Daten möglich machen. Das mitgelieferte Runtime-Modul ermöglicht es Entwicklern, ihre Anwendungen auch denen zugänglich zu machen, die das Superbase-Professional-Entwickler-Paket nicht besitzen.
Hardware-Anforderungen: Amiga 500, 1000, 2000 mit mindestens 512 Kbyte RAM (empfohlen: 1 Mbyte RAM, Festplatte). Lieferbar 2. Quartal 1989, Bestell-Nr. 54136

*Unverbindliche Preisempfehlung

Markt&Technik-Bücher und -Software erhalten Sie bei Ihrem Buchhändler, in Computerfachgeschäften und in den Fachabteilungen der Warenhäuser



228/912

```
374 K9 DATA 23,FC,00,00,00,01,00,00,2E,62
375 XR DATA 2C,79,00,00,3A,46,20,79,00,00
376 kI DATA 3A,56,4E,AE,FF,B8,23,FC,00,00
377 gl DATA 00,00,00,00,2D,A8,2C,79,00,00
378 If DATA 3A,3E,20,79,00,00,3A,62,43,F9
379 oE DATA 00,00,32,6E,30,3C,00,20,4E,AE
380 mB DATA FF,40,2C,79,00,00,3A,42,24,3C
381 vR DATA 00,00,03,ED,22,3C,00,00,2E,EA
382 7m DATA 4E,AE,FF,E2,23,C0,00,00,3A,86
383 N1 DATA 67,00,01,06,22,00,26,3C,00,00
384 hb DATA 00,04,24,3C,00,00,00,00,4E,AE
385 CJ DATA FF,D6,67,00,00,CC,0C,B9,41,4F
386 jK DATA 45,20,00,00,00,00,66,00,00,F6
387 k5 DATA 22,39,00,00,3A,86,26,3C,00,00
388 pv DATA 00,02,24,3C,00,00,00,04,4E,AE
389 cI DATA FF,D6,67,00,00,A4,20,39,00,00
390 sp DATA 2E,62,D0,79,00,00,00,04,B0,B9
391 xu DATA 00,00,3A,78,62,00,00,8E,22,39
392 Iy DATA 00,00,3A,86,26,3C,00,00,00,20
393 Sr DATA 24,3C,00,00,32,8E,4E,AE,FF,D6
394 k3 DATA 67,00,00,74,42,82,24,39,00,00
395 Ad DATA 2E,62,94,BC,00,00,00,01,C4,FC
396 mo DATA 02,80,D4,B9,00,00,3A,4A,42,83
397 51 DATA 36,39,00,00,00,04,C6,FC,02,80
398 xy DATA 22,39,00,00,3A,86,4E,AE,FF,D6
399 1K DATA 67,00,00,42,4E,B9,00,00,2A,F2
400 dl DATA 2C,79,00,00,3A,3E,20,79,00,00
401 ys DATA 3A,62,43,F9,00,00,32,6E,30,3C
402 eV DATA 00,20,4E,AE,FF,40,4E,B9,00,00
403 x5 DATA 27,A6,4E,B9,00,00,05,C2,2C,79
404 sn DATA 00,00,3A,42,22,39,00,00,3A,86
405 cI DATA 4E,AE,FF,DC,60,00,F8,2A,20,79
406 14 DATA 00,00,2D,B4,2C,79,00,00,3A,46
407 pn DATA 4E,AE,FF,A0,2C,79,00,00,3A,42
408 AO DATA 22,39,00,00,3A,86,4E,AE,FF,DC
409 gS DATA 60,00,F8,06,20,79,00,00,2D,B4
410 Ud DATA 2C,79,00,00,3A,46,4E,AE,FF,A0
411 10 DATA 60,00,F7,F2,0C,B9,46,4F,52,4D
412 6F DATA 00,00,00,00,66,BC,22,39,00,00
413 VQ DATA 3A,86,26,3C,00,00,00,04,24,3C
414 gN DATA 00,00,00,00,4E,AE,FF,D6,67,A4
415 Gb DATA 0C,B9,00,00,04,00,00,00,00,00
416 Bk DATA 64,98,22,39,00,00,3A,86,26,3C
417 SE DATA 00,00,04,00,24,3C,00,00,00,00
418 04 DATA 4E,AE,FF,D6,41,F9,00,00,00,00
419 1Y DATA 20,18,B0,BC,49,4C,42,4D,66,00
420 eu DATA FF,72,20,18,B0,BC,42,4D,48,44
421 Iw DATA 67,00,00,3E,B1,FC,00,00,04,00
422 Sn DATA 66,EC,60,00,FF,5A,20,18,B0,BC
423 1j DATA 43,4D,41,50,67,00,00,50,B1,FC
424 Jb DATA 00,00,04,00,66,EC,60,00,FF,42
425 xS DATA 20,18,B0,BC,42,4F,44,59,67,00
426 jH DATA 00,6A,B1,FC,00,00,04,00,66,EC
427 du DATA 60,00,FF,2A,D1,FC,00,00,00,04
428 zz DATA 20,18,B0,BC,00,20,00,20,66,00
429 OE DATA FF,18,D1,FC,00,00,00,04,10,18
430 w1 DATA B0,3C,00,04,66,00,FF,08,D1,FC
431 Hy DATA 00,00,00,03,60,A6,D1,FC,00,00
432 G1 DATA 00,04,43,F9,00,00,32,8E,20,3C
433 Ow DATA 00,00,00,0F,42,41,42,42,12,18
434 34 DATA E9,49,84,41,42,41,12,18,84,41
435 n4 DATA 42,41,12,18,E8,49,84,41,32,C2
436 1m DATA 51,C8,FF,E4,60,8C,D1,FC,00,00
437 I6 DATA 00,04,20,39,00,00,2E,62,90,BC
438 Mk DATA 00,00,00,01,C0,FC,02,80,D0,B9
439 4f DATA 00,00,3A,4A,22,40,24,40,26,40
440 Pn DATA 28,40,D5,FC,00,00,00,80,D7,FC
441 Nv DATA 00,00,01,00,D9,FC,00,00,01,80
442 Th DATA 20,3C,00,00,00,1F,22,D8,24,D8
443 21 DATA 26,D8,28,D8,51,C8,FF,F6,60,00
444 Xh DATA FE,42,2C,79,00,00,3A,46,20,79
445 H8 DATA 00,00,3A,56,4E,AE,FF,B8,20,79
446 q2 DATA 00,00,3A,62,30,3C,00,1F,32,3C
447 EY DATA 00,0D,34,01,36,01,2C,79,00,00
448 AP DATA 3A,3E,4E,AE,FE,E0,20,79,00,00
449 zN DATA 3A,62,30,3C,00,1E,32,3C,00,03
450 Xn DATA 34,01,36,01,4E,AE,FE,E0,2C,79
451 tf DATA 00,00,3A,46,23,FC,00,00,2F,3E
452 yd DATA 00,00,2D,A8,41,F9,00,00,2D,96
453 OC DATA 4E,AE,FF,34,23,C0,00,00,3A,56
454 5p DATA 20,40,23,E8,00,32,00,00,3A,6A
455 Ci DATA 23,E8,00,32,00,00,3A,6E,22,68
456 oE DATA 00,32,30,3C,00,01,2C,79,00,00
457 CW DATA 3A,3E,4E,AE,FE,AA,22,79,00,00
458 Id DATA 3A,6A,30,3C,00,01,4E,AE,FE,A4
459 C9 DATA 22,79,00,00,3A,6A,30,3C,00,00
460 p0 DATA 32,3C,00,00,34,3C,00,DF,36,3C
461 7L DATA 00,54,4E,AE,FE,CE,22,79,00,00
462 5G DATA 3A,6A,30,3C,00,00,32,3C,00,55
463 Ca DATA 34,3C,00,35,36,3C,00,61,4E,AE
464 GP DATA FE,CE,22,79,00,00,3A,6A,30,3C
465 Hk DATA 00,59,32,3C,00,55,34,3C,00,85
466 dJ DATA 36,3C,00,61,4E,AE,FE,CE,22,79
467 b5 DATA 00,00,3A,6A,30,3C,00,A9,32,3C
468 7A DATA 00,55,34,3C,00,DF,36,3C,00,61
469 b1 DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
470 dX DATA 30,3C,00,00,32,3C,00,61,34,3C
471 Mo DATA 00,DF,36,3C,00,72,4E,AE,FE,CE
472 PM DATA 22,79,00,00,3A,6A,30,3C,00,00
473 la DATA 32,3C,00,73,34,3C,00,1D,36,3C
474 EK DATA 00,7F,4E,AE,FE,CE,22,79,00,00
475 tN DATA 3A,6A,30,3C,00,B1,32,3C,00,73
476 Ov DATA 34,3C,00,DF,36,3C,00,7F,4E,AE
477 Tc DATA FE,CE,22,79,00,00,3A,6A,30,3C
478 Fz DATA 00,00,32,3C,00,7F,34,3C,00,DF
479 Y2 DATA 36,3C,00,DF,4E,AE,FE,CE,4B,F9
480 NB DATA 00,00,3B,A5,3A,3C,00,14,3C,3C
481 NI DATA 00,1F,3E,3C,00,4F,4E,B9,00,00
482 bG DATA 04,18,4B,F9,00,00,3B,B9,3A,3C
483 Gc DATA 00,04,3C,3C,00,5F,3E,3C,00,5D
484 ji DATA 4E,B9,00,00,04,18,4B,F9,00,00
485 yS DATA 3B,96,3A,3C,00,09,3C,3C,00,1F
486 OQ DATA 3E,3C,00,6B,4E,B9,00,00,04,18
487 Fs DATA 4B,F9,00,00,3B,9F,3A,3C,00,06
488 4j DATA 3C,3C,00,58,3E,3C,00,91,4E,B9
489 7E DATA 00,00,04,18,30,3C,00,02,38,3C
490 wJ DATA 00,54,3A,3C,00,87,3C,3C,00,8C
491 Ra DATA 3E,3C,00,96,4E,B9,00,00,04,88
492 mE DATA 30,39,00,00,30,02,08,00,00,07
493 Qq DATA 66,00,00,36,08,39,00,06,00,BF
494 JO DATA E0,01,66,E8,22,79,00,00,3A,56
495 L5 DATA 3A,29,00,0E,3C,29,00,0C,BA,7C
496 oR DATA 00,54,65,D4,BC,7C,00,87,65,CE
497 gT DATA BA,7C,00,8C,62,C8,BC,7C,00,96
498 KZ DATA 62,C2,60,00,00,0E,30,39,00,00
499 5N DATA 30,02,08,00,00,07,66,F4,0C,B9
500 u1 DATA 00,00,00,00,00,00,2F,86,66,00
501 12 DATA 00,0C,23,FC,00,00,00,01,00,00
502 aF DATA 2F,86,2C,79,00,00,3A,46,20,79
503 50 DATA 00,00,3A,56,4E,AE,FF,B8,23,FC
504 YO DATA 00,00,00,00,00,00,2D,A8,2C,79
505 X1 DATA 00,00,3A,3E,20,79,00,00,3A,62
506 OU DATA 43,F9,00,00,32,6E,30,3C,00,20
507 9v DATA 4E,AE,FF,40,0C,B9,00,00,00,00
508 1K DATA 00,00,2F,86,66,00,00,0C,23,FC
509 of DATA 00,00,00,01,00,00,2F,86,0C,B9
510 RO DATA 00,00,00,00,00,00,2F,E2,66,00
511 BC DATA 00,0C,23,FC,00,00,00,01,00,00
512 Cy DATA 2F,E2,20,39,00,00,2F,86,22,39
513 UT DATA 00,00,2F,E2,B2,80,6C,00,00,0E
514 xI DATA 23,C1,00,00,2F,86,23,C0,00,00
515 Ng DATA 2F,E2,20,39,00,00,3A,78,B0,B9
516 40 DATA 00,00,2F,E2,65,00,01,04,2C,79
517 eB DATA 00,00,3A,42,24,3C,00,00,03,EE
518 oZ DATA 22,3C,00,00,2E,EA,4E,AE,FF,E2
519 4J DATA 23,C0,00,00,3A,86,67,00,00,E4
520 I7 DATA 23,FC,41,4F,45,20,00,00,00,00
521 wn DATA 22,00,24,3C,00,00,00,00,26,3C
522 k4 DATA 00,00,00,04,4E,AE,FF,D0,67,00
523 g2 DATA 00,A0,20,39,00,00,2F,E2,90,B9
524 3C DATA 00,00,2F,86,D0,7C,00,01,33,C0
525 8x DATA 00,00,00,00,22,39,00,00,3A,86
526 w7 DATA 24,3C,00,00,00,00,26,3C,00,00
527 1a DATA 00,02,4E,AE,FF,D0,67,00,00,70
528 pI DATA 22,39,00,00,3A,86,24,3C,00,00
529 kp DATA 32,8E,26,3C,00,00,00,20,4E,AE
530 w1 DATA FF,D0,67,00,00,56,4E,B9,00,00
531 Mw DATA 2B,20,24,39,00,00,2F,86,94,BC
532 Kq DATA 00,00,00,01,C4,FC,02,80,D4,B9
533 Lr DATA 00,00,3A,4A,26,39,00,00,2F,E2
534 Gc DATA 96,B9,00,00,2F,86,D6,BC,00,00
535 H1 DATA 00,01,C6,FC,02,80,22,39,00,00
536 Dy DATA 3A,86,4E,AE,FF,D0,67,00,00,16
537 Wg DATA 4E,B9,00,00,2A,F2,22,39,00,00
538 2V DATA 3A,86,4E,AE,FF,DC,60,00,F2,F6
539 dE DATA 20,79,00,00,2D,B4,2C,79,00,00
540 41 DATA 3A,46,4E,AE,FF,A0,2C,79,00,00
541 Ut DATA 3A,42,22,39,00,00,3A,86,4E,AE
542 fr DATA FF,DC,60,00,F2,D2,20,79,00,00
543 Fj DATA 2D,B4,2C,79,00,00,3A,46,4E,AE
544 Jj DATA FF,A0,60,00,F2,BE,2C,79,00,00
545 Y7 DATA 3A,3E,43,F9,00,00,3A,98,20,3C
546 mp DATA 00,00,00,10,4E,AE,FE,AA,43,F9
547 ze DATA 00,00,3A,98,42,80,42,81,34,3C
548 qD DATA 00,1F,36,3C,00,1F,4E,AE,FE,CE
549 hY DATA 4E,B9,00,00,27,A6,60,00,F2,88
550 Pm DATA 20,79,00,00,3B,10,61,00,00,28
551 bE DATA 20,79,00,00,3B,14,61,00,00,1E
552 Rr DATA 20,79,00,00,3B,18,61,00,00,14
553 y1 DATA 20,79,00,00,3B,1C,61,00,00,0A
554 ab DATA 61,00,12,D8,60,00,F2,58,20,3C
555 k8 DATA 00,00,00,1F,22,10,24,10,48,42
556 xg DATA C2,BC,00,00,FF,FF,C4,BC,00,00
557 I9 DATA FF,FF,42,83,42,84,2E,3C,00,00
558 zR DATA 00,0F,0F,01,67,00,00,10,46,47
559 La DATA CE,7C,00,0F,0F,C3,46,47,CE,7C
560 Hh DATA 00,0F,51,CF,FF,EA,2E,3C,00,00
561 4X DATA 00,0F,0F,02,67,00,00,10,46,47
562 Rh DATA CE,7C,00,0F,0F,C4,46,47,CE,7C
563 3s DATA 00,0F,51,CF,FF,EA,48,43,86,84
564 oK DATA 20,C3,51,C8,FF,A6,4E,75,20,79
565 yu DATA 00,00,3B,10,61,00,00,28,20,79
566 u6 DATA 00,00,3B,14,61,00,00,1E,20,79
567 xw DATA 00,00,3B,18,61,00,00,14,20,79
568 MD DATA 00,00,3B,1C,61,00,00,0A,61,00
569 f8 DATA 12,44,60,00,F1,C4,22,48,D3,FC
570 tN DATA 00,00,00,80,20,3C,00,00,00,0F
571 EG DATA 22,10,24,21,22,81,20,C2,51,C8
572 Tq DATA FF,F6,4E,75,20,79,00,00,3B,10
573 zp DATA 61,00,00,28,20,79,00,00,3B,14
574 ep DATA 61,00,00,1E,20,79,00,00,3B,18
575 05 DATA 61,00,00,14,20,79,00,00,3B,1C
576 th DATA 61,00,00,0A,61,00,11,F8,60,00
577 Rd DATA F1,78,43,F9,00,00,00,00,24,48
578 3s DATA 2C,3C,00,00,00,1F,2E,3C,00,00
579 CY DATA 00,1F,42,83,20,4A,20,18,0D,00
580 Ae DATA 67,00,00,10,46,47,CE,7C,00,1F
581 V2 DATA 0F,C3,46,47,CE,7C,00,1F,51,CF
582 qw DATA FF,E8,22,C3,51,CE,FF,D8,43,F9
583 re DATA 00,00,00,00,20,4A,2C,3C,00,00
584 6S DATA 00,1F,20,D9,51,CE,FF,FC,4E,75
585 FI DATA 20,79,00,00,3A,62,30,3C,00,1F
586 oX DATA 32,3C,00,0D,34,01,36,01,2C,79
587 Ob DATA 00,00,3A,3E,4E,AE,FE,E0,20,79
588 4F DATA 00,00,3A,62,30,3C,00,1E,32,3C
589 ea DATA 00,03,34,01,36,01,4E,AE,FE,E0
590 fm DATA 2C,79,00,00,3A,46,23,FC,00,00
591 T2 DATA 31,5A,00,00,2D,A8,41,F9,00,00
592 cb DATA 2D,96,4E,AE,FF,34,23,C0,00,00
593 fL DATA 3A,56,20,40,23,E8,00,32,00,00
594 KM DATA 3A,6A,23,E8,00,32,00,00,3A,6E
595 dE DATA 22,68,00,32,30,3C,00,01,2C,79
596 Vx DATA 00,00,3A,3E,4E,AE,FE,AA,22,79
597 81 DATA 00,00,3A,6A,30,3C,00,01,4E,AE
598 cp DATA FE,A4,22,79,00,00,3A,6A,30,3C
599 uB DATA 00,00,32,3C,00,00,34,3C,00,DF
600 na DATA 36,3C,00,54,4E,AE,FE,CE,22,79
601 MQ DATA 00,00,3A,6A,30,3C,00,00,32,3C
602 uP DATA 00,55,34,3C,00,35,36,3C,00,61
603 lv DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
604 mY DATA 30,3C,00,59,32,3C,00,55,34,3C
605 p9 DATA 00,85,36,3C,00,61,4E,AE,FE,CE
606 01 DATA 22,79,00,00,3A,6A,30,3C,00,A9
607 qA DATA 32,3C,00,55,34,3C,00,DF,36,3C
608 H1 DATA 00,61,4E,AE,FE,CE,22,79,00,00
609 CK DATA 3A,6A,30,3C,00,00,32,3C,00,61
610 Dt DATA 34,3C,00,DF,36,3C,00,68,4E,AE
611 dm DATA FE,CE,22,79,00,00,3A,6A,30,3C
612 rx DATA 00,00,32,3C,00,69,34,3C,00,79
613 Eu DATA 36,3C,00,75,4E,AE,FE,CE,22,79
614 Ip DATA 00,00,3A,6A,30,3C,00,9D,32,3C
615 LE DATA 00,69,34,3C,00,DF,36,3C,00,75
616 y8 DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
617 JI DATA 30,3C,00,00,32,3C,00,75,34,3C
618 76 DATA 00,DF,36,3C,00,DF,4E,AE,FE,CE
619 7e DATA 4B,F9,00,00,3C,4F,3A,3C,00,13
620 rt DATA 3C,3C,00,23,3E,3C,00,4F,4E,B9
621 F1 DATA 00,00,04,18,4B,F9,00,00,3B,B9
622 GN DATA 3A,3C,00,04,3C,3C,00,5F,3E,3C
623 2t DATA 00,5D,4E,B9,00,00,04,18,4B,F9
624 gB DATA 00,00,3C,62,3A,3C,00,06,3C,3C
625 wb DATA 00,3F,3E,3C,00,71,4E,B9,00,00
```

```
626 K3  DATA 04,18,4B,F9,00,00,3B,9F,3A,3C
627 pa  DATA 00,06,3C,3C,00,58,3E,3C,00,91
628 78  DATA 4E,B9,00,00,04,18,30,3C,00,02
629 N1  DATA 38,3C,00,54,3A,3C,00,87,3C,3C
630 RU  DATA 00,8C,3E,3C,00,96,4E,B9,00,00
631 cx  DATA 04,88,30,39,00,00,32,1E,08,00
632 vq  DATA 00,07,66,00,00,36,08,39,00,06
633 3M  DATA 00,BF,E0,01,66,E8,22,79,00,00
634 AQ  DATA 3A,56,3A,29,00,0E,3C,29,00,0C
635 Ey  DATA BA,7C,00,54,65,D4,BC,7C,00,87
636 9M  DATA 65,CE,BA,7C,00,8C,62,C8,BC,7C
637 rF  DATA 00,96,62,C2,60,00,00,0E,30,39
638 xw  DATA 00,00,32,1E,08,00,00,07,66,F4
639 nh  DATA 2C,79,00,00,3A,46,20,79,00,00
640 OY  DATA 3A,56,4E,AE,FF,B8,23,FC,00,00
641 w1  DATA 00,00,00,00,2D,A8,2C,79,00,00
642 Yv  DATA 3A,3E,20,79,00,00,3A,62,43,F9
643 4U  DATA 00,00,32,6E,30,3C,00,20,4E,AE
644 k2  DATA FF,40,4A,B9,00,00,31,A2,62,00
645 LM  DATA 00,0C,23,FC,00,00,00,01,00,00
646 ry  DATA 31,A2,4A,B9,00,00,31,FE,62,00
647 NO  DATA 00,0C,23,FC,00,00,00,01,00,00
648 B1  DATA 31,FE,4A,B9,00,00,32,5A,63,00
649 J0  DATA 01,00,20,39,00,00,31,A2,D0,B9
650 x0  DATA 00,00,32,5A,90,BC,00,00,00,01
651 Jm  DATA B0,B9,00,00,3A,78,62,00,00,EE
652 3E  DATA 22,39,00,00,31,FE,D2,B9,00,00
653 SQ  DATA 32,5A,92,BC,00,00,00,01,B2,B9
654 hR  DATA 00,00,3A,78,62,00,00,D2,B2,80
655 c8  DATA 67,00,00,C2,B0,B9,00,00,31,FE
656 aR  DATA 62,00,00,4C,20,39,00,00,32,5A
657 dz  DATA C0,FC,00,A0,90,BC,00,00,00,01
658 pb  DATA 22,39,00,00,31,A2,92,BC,00,00
659 CP  DATA 00,01,C2,FC,02,80,D2,B9,00,00
660 JQ  DATA 3A,4A,20,41,22,39,00,00,31,FE
661 zV  DATA 92,BC,00,00,00,01,C2,FC,02,80
662 40  DATA D2,B9,00,00,3A,4A,22,41,22,D8
663 av  DATA 51,C8,FF,FC,60,00,00,6E,20,39
664 Fk  DATA 00,00,31,A2,B0,B9,00,00,31,FE
665 PF  DATA 62,A8,20,39,00,00,32,5A,C0,FC
666 95  DATA 00,A0,90,BC,00,00,00,01,22,39
667 CR  DATA 00,00,31,A2,D2,B9,00,00,32,5A
668 6c  DATA 92,BC,00,00,00,01,C2,FC,02,80
669 yA  DATA D2,B9,00,00,3A,4A,20,41,22,39
670 LZ  DATA 00,00,31,FE,D2,B9,00,00,32,5A
671 9f  DATA 92,BC,00,00,00,01,C2,FC,02,80
672 xJ  DATA D2,B9,00,00,3A,4A,22,41,D1,FC
673 Tt  DATA 00,00,00,04,D3,FC,00,00,00,04
674 HA  DATA 23,20,51,C8,FF,FC,4E,B9,00,00
675 yM  DATA 27,A6,60,00,ED,A0,2C,79,00,00
676 00  DATA 3A,46,20,79,00,00,2D,84,4E,AE
677 ZV  DATA FF,A0,60,00,ED,8C,41,F9,00,00
678 t6  DATA 2D,96,2C,79,00,00,3A,46,4E,AE
679 FW  DATA FF,34,4A,80,67,00,ED,76,23,C0
680 UK  DATA 00,00,3A,56,20,40,23,E8,00,32
681 WW  DATA 00,00,3A,6A,22,68,00,32,30,3C
682 dt  DATA 00,01,2C,79,00,00,3A,3E,4E,AE
683 Fc  DATA FE,AA,22,79,00,00,3A,6A,30,3C
684 8A  DATA 00,01,4E,AE,FE,A4,22,79,00,00
685 Tt  DATA 3A,6A,42,40,42,41,34,3C,00,DF
686 Da  DATA 36,3C,00,DF,4E,AE,FE,CE,23,F9
687 6U  DATA 00,00,3A,6A,00,00,3A,6E,4B,F9
688 ER  DATA 00,00,3B,BD,3A,3C,00,0C,3C,3C
689 or  DATA 00,40,3E,3C,00,50,4E,B9,00,00
690 0g  DATA 04,18,4B,F9,00,00,3B,C9,3A,3C
691 pv  DATA 00,0B,3C,3C,00,44,3E,3C,00,5A
692 54  DATA 4E,B9,00,00,04,18,4B,F9,00,00
693 K4  DATA 3B,D4,3A,3C,00,0F,3C,3C,00,34
694 4L  DATA 3E,3C,00,69,4E,B9,00,00,04,18
695 M8  DATA 4B,F9,00,00,3B,E3,3A,3C,00,0F
696 8r  DATA 3C,3C,00,34,3E,3C,00,73,4E,B9
697 At  DATA 00,00,04,18,4B,F9,00,00,3B,F2
698 93  DATA 3A,3C,00,11,3C,3C,00,2C,3E,3C
699 1p  DATA 00,82,4E,B9,00,00,04,18,4B,F9
700 m0  DATA 00,00,3B,9F,3A,3C,00,06,3C,3C
701 Hz  DATA 00,58,3E,3C,00,96,4E,B9,00,00
702 Yb  DATA 04,18,30,3C,00,02,38,3C,00,54
703 mc  DATA 3A,3C,00,8C,3C,3C,00,8C,3E,3C
704 sp  DATA 00,9B,4E,B9,00,00,04,88,08,39
705 1V  DATA 00,06,00,BF,E0,01,66,F6,22,79
706 Uy  DATA 00,00,3A,56,3A,29,00,0E,3C,29
707 Jc  DATA 00,0C,BA,7C,00,54,65,E2,BC,7C
708 mc  DATA 00,8C,65,DC,BA,7C,00,8C,62,D6
709 us  DATA BC,7C,00,9B,62,D0,2C,79,00,00
710 iW  DATA 3A,46,20,79,00,00,3A,56,4E,AE
711 CM  DATA FF,B8,60,00,EC,38,41,F9,00,00
712 3k  DATA 2D,96,23,FC,00,00,30,A2,00,00
713 IV  DATA 2D,A8,2C,79,00,00,3A,46,4E,AE
714 1h  DATA FF,34,23,C0,00,00,3A,56,20,40
715 8q  DATA 23,E8,00,32,00,00,3A,6A,23,E8
716 HQ  DATA 00,32,00,00,3A,6E,22,68,00,32
717 p2  DATA 30,3C,00,01,2C,79,00,00,3A,3E
718 IW  DATA 4E,AE,FE,AA,22,79,00,00,3A,6A
719 aD  DATA 30,3C,00,01,4E,AE,FE,A4,20,79
720 GS  DATA 00,00,3A,62,30,3C,00,1F,32,3C
721 ey  DATA 00,0D,34,01,36,01,2C,79,00,00
722 ap  DATA 3A,3E,4E,AE,FE,E0,20,79,00,00
723 Pn  DATA 3A,62,30,3C,00,1E,32,3C,00,03
724 PO  DATA 34,01,36,01,4E,AE,FE,E0,22,79
725 MQ  DATA 00,00,3A,6A,30,3C,00,00,32,3C
726 wX  DATA 00,00,34,3C,00,DF,36,3C,00,5E
727 1v  DATA 4E,AE,FE,CE,22,79,00,00,3A,6A
728 6K  DATA 30,3C,00,00,32,3C,00,5F,34,3C
729 F1  DATA 00,35,36,3C,00,6B,4E,AE,FE,CE
730 cn  DATA 22,79,00,00,3A,6A,30,3C,00,59
731 iq  DATA 32,3C,00,5F,34,3C,00,85,36,3C
732 vG  DATA 00,6B,4E,AE,FE,CE,22,79,00,00
733 og  DATA 3A,6A,30,3C,00,A9,32,3C,00,5F
734 rh  DATA 34,3C,00,DF,36,3C,00,6B,4E,AE
735 dm  DATA FE,CE,22,79,00,00,3A,6A,30,3C
736 Bq  DATA 00,00,32,3C,00,6B,34,3C,00,DF
737 iC  DATA 36,3C,00,DF,4E,AE,FE,CE,4B,F9
738 RU  DATA 00,00,3C,1A,3A,3C,00,14,3C,3C
739 wf  DATA 00,1F,3E,3C,00,59,4E,B9,00,00
740 1Q  DATA 04,18,4B,F9,00,00,3B,B9,3A,3C
741 MV  DATA 00,04,3C,3C,00,5F,3E,3C,00,66
742 ts  DATA 4E,B9,00,00,04,18,4B,F9,00,00
743 fu  DATA 3B,9F,3A,3C,00,06,3C,3C,00,58
744 Ki  DATA 3E,3C,00,7D,4E,B9,00,00,04,18
745 BS  DATA 30,3C,00,02,38,3C,00,54,3A,3C
746 fx  DATA 00,73,3C,3C,00,8C,3E,3C,00,82
747 Tw  DATA 4E,B9,00,00,04,88,08,39,00,06
748 q8  DATA 00,BF,E0,01,66,F6,22,79,00,00
749 1H  DATA 3A,56,3A,29,00,0E,3C,29,00,0C
750 dH  DATA BA,7C,00,54,65,E2,BC,7C,00,73
751 o9  DATA 65,DC,BA,7C,00,8C,62,D6,BC,7C
752 EA  DATA 00,82,62,D0,2C,79,00,00,3A,46
753 qq  DATA 20,79,00,00,3A,56,4E,AE,FF,B8
754 Q7  DATA 0C,B9,00,00,00,00,00,00,30,EA
755 Ms  DATA 66,00,00,0C,23,FC,00,00,00,01
756 WI  DATA 00,00,30,EA,20,39,00,00,3A,78
757 1D  DATA B0,B9,00,00,30,EA,62,00,00,0C
758 HN  DATA 23,F9,00,00,3A,78,00,00,30,EA
759 Yo  DATA 0C,B9,00,00,00,00,00,00,31,46
760 Rx  DATA 66,00,00,0C,23,FC,00,00,00,01
761 fN  DATA 00,00,31,46,20,39,00,00,3A,78
762 6r  DATA B0,B9,00,00,31,46,62,00,00,0C
763 P4  DATA 23,F9,00,00,3A,78,00,00,31,46
764 Hn  DATA 20,39,00,00,30,EA,22,39,00,00
765 pN  DATA 31,46,B2,80,64,00,00,0E,23,C1
766 8j  DATA 00,00,30,EA,23,C0,00,00,31,46
767 yI  DATA 23,F9,00,00,30,EA,00,00,3A,8E
768 HU  DATA 41,F9,00,00,2D,96,23,FC,00,00
769 05  DATA 00,00,00,00,2D,A8,2C,79,00,00
770 yq  DATA 3A,46,4E,AE,FF,34,23,C0,00,00
771 XD  DATA 3A,56,20,40,23,E8,00,32,00,00
772 CE  DATA 3A,6A,23,E8,00,32,00,00,3A,6E
773 nJ  DATA 22,68,00,32,22,79,00,00,3A,6A
774 kx  DATA 30,3C,00,01,2C,79,00,00,3A,3E
775 qn  DATA 4E,AE,FE,AA,20,79,00,00,3A,62
776 kq  DATA 43,F9,00,00,32,6E,30,3C,00,20
777 VJ  DATA 4E,AE,FF,40,22,79,00,00,3A,6A
778 bG  DATA 30,3C,00,01,4E,AE,FE,A4,22,79
779 EI  DATA 00,00,3A,6A,30,3C,00,00,32,3C
780 r1  DATA 00,00,34,3C,00,DF,36,3C,00,DF
781 zD  DATA 4E,AE,FE,CE,2C,79,00,00,3A,46
782 xq  DATA 20,79,00,00,3A,6A,43,F9,00,00
783 12  DATA 35,9E,30,3C,00,58,32,3C,00,57
784 BM  DATA 4E,AE,FF,8E,2C,79,00,00,3A,3E
785 9v  DATA 4B,F9,00,00,3C,2E,3A,3C,00,0F
786 fe  DATA 3C,3C,00,34,3E,3C,00,4F,4E,B9
787 OE  DATA 00,00,04,18,4B,F9,00,00,3B,9F
788 61  DATA 3A,3C,00,06,3C,3C,00,58,3E,3C
789 s1  DATA 00,7D,4E,B9,00,00,04,18,4B,F9
790 SO  DATA 00,00,3C,3D,3A,3C,00,03,3C,3C
791 s5  DATA 00,63,3E,3C,00,5D,4E,B9,00,00
792 03  DATA 04,18,30,3C,00,02,38,3C,00,54
793 cG  DATA 3A,3C,00,73,3C,3C,00,8C,3E,3C
794 XL  DATA 00,82,4E,B9,00,00,04,88,30,3C
795 EH  DATA 00,10,22,79,00,00,3A,6A,4E,AE
796 4R  DATA FE,AA,22,79,00,00,3A,6A,30,3C
797 wj  DATA 00,5F,32,3C,00,14,34,3C,00,7E
798 LF  DATA 36,3C,00,33,4E,AE,FE,CE,30,3C
799 Ob  DATA 00,02,38,3C,00,5E,3A,3C,00,13
800 eM  DATA 3C,3C,00,7F,3E,3C,00,34,4E,B9
801 tZ  DATA 00,00,04,88,30,3C,00,03,38,3C
802 WM  DATA 00,5D,3A,3C,00,12,3C,3C,00,80
803 rN  DATA 3E,3C,00,35,4E,B9,00,00,04,88
804 Ai  DATA 30,3C,00,04,38,3C,00,5C,3A,3C
805 45  DATA 00,11,3C,3C,00,81,3E,3C,00,36
806 XZ  DATA 4E,B9,00,00,04,88,23,F9,00,00
807 A7  DATA 3B,10,00,00,00,00,23,F9,00,00
808 1W  DATA 3B,14,00,00,00,04,23,F9,00,00
809 Mv  DATA 3B,18,00,00,00,08,23,F9,00,00
810 y0  DATA 3B,1C,00,00,00,0C,23,F9,00,00
811 LK  DATA 3B,20,00,00,00,10,23,F9,00,00
812 1N  DATA 3A,92,00,00,3A,8A,2A,7C,00,DF
813 45  DATA F0,00,23,F8,00,6C,00,00,27,68
814 FM  DATA 21,FC,00,00,26,AC,00,6C,0C,79
815 6i  DATA 00,01,00,00,3A,96,66,00,00,54
816 JT  DATA 20,39,00,DF,F0,04,C0,BC,00,01
817 tf  DATA FF,00,B0,BC,00,01,00,00,66,EC
818 Mh  DATA 2C,79,00,00,3A,3E,41,F9,00,00
819 4G  DATA 3A,98,22,79,00,00,3A,6A,42,80
820 Ec  DATA 42,81,24,3C,00,00,00,5F,26,3C
821 OC  DATA 00,00,00,14,28,3C,00,00,00,20
822 fF  DATA 2A,3C,00,00,00,20,2C,3C,00,00
823 B1  DATA 00,C0,4E,AE,FD,D8,42,79,00,00
824 J8  DATA 3A,96,08,39,00,06,00,BF,E0,01
825 hv  DATA 66,98,22,79,00,00,3A,56,3A,29
826 1A  DATA 00,0E,3C,29,00,0C,BA,7C,00,58
827 pR  DATA 65,00,00,1E,BC,7C,00,57,65,00
828 sv  DATA 00,16,BA,7C,00,5F,62,00,00,0E
829 xY  DATA BC,7C,00,5E,62,00,00,06,60,00
830 Z6  DATA 0A,06,BA,7C,00,80,65,00,00,22
831 E1  DATA BC,7C,00,57,65,00,00,1A,BA,7C
832 AB  DATA 00,87,62,00,00,12,BC,7C,00,5E
833 7S  DATA 62,00,00,0A,60,00,0A,22,60,00
834 oI  DATA FF,40,BA,7C,00,54,65,00,FF,38
835 GI  DATA BC,7C,00,73,65,00,FF,30,BA,7C
836 ZN  DATA 00,8C,62,00,FF,28,BC,7C,00,82
837 da  DATA 62,00,FF,20,2A,7C,00,DF,F0,00
838 Vm  DATA 21,F9,00,00,27,68,00,6C,23,F9
839 0a  DATA 00,00,00,00,00,00,3B,10,23,F9
840 P7  DATA 00,00,00,04,00,00,3B,14,23,F9
841 oe  DATA 00,00,00,08,00,00,3B,18,23,F9
842 t5  DATA 00,00,00,0C,00,00,3B,1C,23,F9
843 1J  DATA 00,00,00,10,00,00,3B,20,2C,79
844 sq  DATA 00,00,3A,46,20,79,00,00,3A,56
845 t0  DATA 4E,AE,FF,B8,23,F9,00,00,3A,66
846 9Q  DATA 00,00,3A,6E,23,FC,00,00,00,00
847 EM  DATA 00,00,2D,A8,60,00,E6,E6,30,39
848 wb  DATA 00,00,3A,76,B0,7C,00,01,66,00
849 53  DATA 00,06,60,00,E6,D4,04,B9,00,00
850 yv  DATA 02,80,00,00,3B,10,04,B9,00,00
851 BC  DATA 02,80,00,00,3B,14,04,B9,00,00
852 OT  DATA 02,80,00,00,3B,18,04,B9,00,00
853 wC  DATA 02,80,00,00,3B,1C,04,B9,00,00
854 cQ  DATA 02,80,00,00,3B,20,04,79,00,01
855 2L  DATA 00,00,3A,76,4E,B9,00,00,2A,42
856 PX  DATA 4E,B9,00,00,27,A6,60,00,E6,8A
857 13  DATA 30,39,00,00,3A,76,B0,B9,00,00
858 iE  DATA 3A,78,66,00,00,06,60,00,E6,76
859 pA  DATA 06,B9,00,00,02,80,00,00,3B,10
860 AZ  DATA 06,B9,00,00,02,80,00,00,3B,14
861 Vy  DATA 06,B9,00,00,02,80,00,00,3B,18
862 P3  DATA 06,B9,00,00,02,80,00,00,3B,1C
863 xJ  DATA 06,B9,00,00,02,80,00,00,3B,20
864 Gq  DATA 06,79,00,01,00,00,3A,76,4E,B9
865 Z6  DATA 00,00,2A,42,4E,B9,00,00,27,A6
866 1S  DATA 60,00,E6,2C,42,80,30,39,00,00
867 hy  DATA 3A,72,C0,FC,00,02,20,40,D1,FC
868 14  DATA 00,00,32,8E,30,10,C0,7C,0F,00
869 3U  DATA E0,58,D0,7C,00,01,C0,7C,00,0F
870 GW  DATA E1,58,02,50,00,FF,81,50,4E,B9
871 Jt  DATA 00,00,27,6C,4E,B9,00,00,05,C2
872 NC  DATA 60,00,E5,F0,42,80,30,39,00,00
873 n4  DATA 3A,72,C0,FC,00,02,20,40,D1,FC
874 rA  DATA 00,00,32,8E,30,10,C0,7C,0F,00
```

Listing 1. (Fortsetzung)

```
875 cs  DATA E0,58,90,7C,00,01,C0,7C,00,0F
876.Mc  DATA E1,58,02,50,00,FF,81,50,4E,B9
877 Pz  DATA 00,00,27,6C,4E,B9,00,00,05,C2
878 DM  DATA 60,00,E5,B4,42,80,30,39,00,00
879 tA  DATA 3A,72,C0,FC,00,02,20,40,D1,FC
880 fy  DATA 00,00,32,8E,30,10,C0,7C,00,F0
881 dC  DATA E8,58,D0,7C,00,01,C0,7C,00,0F
882 uI  DATA E9,58,02,50,0F,0F,81,50,4E,B9
883 V5  DATA 00,00,27,6C,4E,B9,00,00,05,C2
884 NI  DATA 60,00,E5,78,42,80,30,39,00,00
885 zG  DATA 3A,72,C0,FC,00,02,20,40,D1,FC
886 14  DATA 00,00,32,8E,30,10,C0,7C,00,F0
887 Ca  DATA E8,58,90,7C,00,01,C0,7C,00,0F
888 00  DATA E9,58,02,50,0F,0F,81,50,4E,B9
889 bB  DATA 00,00,27,6C,4E,B9,00,00,05,C2
890 Rn  DATA 60,00,E5,3C,42,80,30,39,00,00
891 5M  DATA 3A,72,C0,FC,00,02,20,40,D1,FC
892 DW  DATA 00,00,32,8E,30,10,C0,7C,00,0F
893 fL  DATA D0,7C,00,01,C0,7C,00,0F,02,50
894 VW  DATA 0F,F0,81,50,4E,B9,00,00,27,6C
895 1h  DATA 4E,B9,00,00,05,C2,60,00,E5,04
896 CD  DATA 42,80,30,39,00,00,3A,72,C0,FC
897 iv  DATA 00,02,20,40,D1,FC,00,00,32,8E
898 JJ  DATA 30,10,C0,7C,00,0F,90,7C,00,01
899 sJ  DATA C0,7C,00,0F,02,50,0F,F0,81,50
900 bf  DATA 4E,B9,00,00,27,6C,4E,B9,00,00
901 vc  DATA 05,C2,60,00,E4,CC,2C,79,00,00
902 JD  DATA 3A,46,41,F9,00,00,2D,C6,4E,AE
903 4k  DATA FF,34,23,C0,00,00,3A,56,20,40
904 Bt  DATA 23,E8,00,32,00,00,3A,6A,23,E8
905 KT  DATA 00,32,00,00,3A,6E,22,68,00,32
906 s5  DATA 30,3C,00,01,2C,79,00,00,3A,3E
907 yv  DATA 4E,AE,FE,AA,20,79,00,00,3A,62
908 nr  DATA 30,3C,00,1F,32,3C,00,0D,34,01
909 tC  DATA 36,01,4E,AE,FE,E0,20,79,00,00
910 Qo  DATA 3A,62,30,3C,00,1E,32,3C,00,03
911 QP  DATA 34,01,36,01,4E,AE,FE,E0,22,79
912 Dn  DATA 00,00,3A,6A,30,3C,00,01,4E,AE
913 OH  DATA FE,A4,22,79,00,00,3A,6A,42,40
914 k8  DATA 42,41,34,3C,00,8F,36,3C,00,0C
915 DM  DATA 4E,AE,FE,CE,42,40,32,3C,00,0D
916 KE  DATA 34,3C,00,36,36,3C,00,18,22,79
917 m4  DATA 00,00,3A,6A,4E,AE,FE,CE,30,3C
918 Ig  DATA 00,5A,32,3C,00,0D,34,3C,00,8F
919 CG  DATA 36,3C,00,18,22,79,00,00,3A,6A
920 TS  DATA 4E,AE,FE,CE,42,40,32,3C,00,19
921 AC  DATA 34,3C,00,8F,36,3C,00,2C,22,79
922 r9  DATA 00,00,3A,6A,4E,AE,FE,CE,30,3C
923 YQ  DATA 00,03,42,44,42,45,3C,3C,00,8F
924 Ay  DATA 3E,3C,00,2C,4E,B9,00,00,04,88
925 ls  DATA 4B,F9,00,00,3C,03,3A,3C,00,10
926 fc  DATA 3C,3C,00,08,3E,3C,00,08,4E,B9
927 eU  DATA 00,00,04,18,4B,F9,00,00,3B,9F
928 kV  DATA 3A,3C,00,06,3C,3C,00,30,3E,3C
929 Ba  DATA 00,25,4E,B9,00,00,04,18,30,3C
930 Wt  DATA 00,02,38,3C,00,2C,3A,3C,00,1B
931 sY  DATA 3C,3C,00,64,3E,3C,00,2A,4E,B9
932 pv  DATA 00,00,04,88,30,39,00,00,30,52
933 FP  DATA 08,00,00;07,66,00,00,36,08,39
934 XI  DATA 00,06,00,BF,E0,01,66,E8,22,79
935 Bf  DATA 00,00,3A,56,3A,29,00,0E,3C,29
936 iE  DATA 00,0C,BA,7C,00,2C,65,D4,BC,7C
937 oR  DATA 00,1B,65,CE,BA,7C,00,64,62,C8
938 D0  DATA BC,7C,00,2A,62,C2,60,00,00,0E
939 9g  DATA 30,39,00,00,30,52,08,00,00,07
940 wl  DATA 66,F4,2C,79,00,00,3A,46,20,79
941 Ps  DATA 00,00,3A,56,4E,AE,FF,B8,0C,B9
942 xR  DATA 00,00,00,00,00,00,30,8E,66,00
943 9A  DATA 00,0C,23,FC,00,00,00,01,00,00
944 3a  DATA 30,8E,20,39,00,00,3A,78,B0,B9
945 DE  DATA 00,00,30,8E,64,00,00,0C,23,F9
946 XN  DATA 00,00,3A,78,00,00,30,8E,20,39
947 hW  DATA 00,00,30,8E,33,C0,00,00,3A,76
948 Kw  DATA 90,BC,00,00,00,01,C0,FC,02,80
949 AG  DATA D0,B9,00,00,3A,4A,23,C0,00,00
950 b2  DATA 3B,10,D0,BC,00,00,00,80,23,C0
951 Bp  DATA 00,00,3B,14,D0,BC,00,00,00,80
952 sv  DATA 23,C0,00,00,3B,18,D0,BC,00,00
953 2n  DATA 00,80,23,C0,00,00,3B,1C,D0,BC
954 9P  DATA 00,00,00,80,23,C0,00,00,3B,20
955 ai  DATA 2C,79,00,00,3A,3E,20,79,00,00
956 vp  DATA 3A,62,43,F9,00,00,32,6E,30,3C
957 bS  DATA 00,20,4E,AE,FF,40,4E,B9,00,00
958 4W  DATA 2A,42,4E,B9,00,00,27,A6,60,00
959 Jy  DATA E2,8C,23,FC,00,00,00,64,00,00
960 1K  DATA 30,8E,20,3C,00,31,30,30,E1,88
961 OT  DATA 23,C0,00,00,30,96,23,C0,00,00
962 V8  DATA 30,9C,2C,79,00,00,3A,46,41,F9
963 I4  DATA 00,00,2D,C6,4E,AE,FF,34,23,C0
964 4u  DATA 00,00,3A,56,20,40,23,E8,00,32
965 ek  DATA 00,00,3A,6A,23,E8,00,32,00,00
966 yb  DATA 3A,6E,22,68,00,32,30,3C,00,01
967 V8  DATA 2C,79,00,00,3A,3E,4E,AE,FE,AA
968 QT  DATA 20,79,00,00,3A,62,30,3C,00,1F
969 rq  DATA 32,3C,00,0D,34,01,36,01,4E,AE
970 Si  DATA FE,E0,20,79,00,00,3A,62,30,3C
971 MF  DATA 00,1E,32,3C,00,03,34,01,36,01
972 Ev  DATA 4E,AE,FE,E0,22,79,00,00,3A,6A
973 kP  DATA 30,3C,00,01,4E,AE,FE,A4,22,79
974 YQ  DATA 00,00,3A,6A,42,40,42,41,34,3C
975 7X  DATA 00,8F,36,3C,00,0C,4E,AE,FE,CE
976 7L  DATA 42,40,32,3C,00,0D,34,3C,00,36
977 8C  DATA 36,3C,00,18,22,79,00,00,3A,6A
978 tE  DATA 4E,AE,FE,CE,30,3C,00,5A,32,3C
979 G6  DATA 00,0D,34,3C,00,8F,36,3C,00,18
980 Fg  DATA 22,79,00,00,3A,6A,4E,AE,FE,CE
981 Ji  DATA 42,40,32,3C,00,19,34,3C,00,8F
982 fq  DATA 36,3C,00,2C,22,79,00,00,3A,6A
983 mW  DATA 4E,AE,FE,CE,30,3C,00,03,42,44
984 oF  DATA 42,45,3C,3C,00,8F,3E,3C,00,2C
985 28  DATA 4E,B9,00,00,04,88,4B,F9,00,00
986 RO  DATA 3C,40,3A,3C,00,0F,3C,3C,00,0C
987 Bp  DATA 3E,3C,00,08,4E,B9,00,00,04,18
988 Kx  DATA 4B,F9,00,00,3B,9F,3A,3C,00,06
989 cE  DATA 3C,3C,00,30,3E,3C,00,25,4E,B9
990 CJ  DATA 00,00,04,18,30,3C,00,02,38,3C
991 CL  DATA 00,2C,3A,3C,00,1B,3C,3C,00,64
992 Cy  DATA 3E,3C,00,2A,4E,B9,00,00,04,88
993 1Y  DATA 30,39,00,00,30,52,08,00,00,07
994 Vv  DATA 66,00,00,36,08,39,00,06,00,BF
995 OT  DATA E0,01,66,E8,22,79,00,00,3A,56
996 QA  DATA 3A,29,00,0E,3C,29,00,0C,BA,7C
997 T9  DATA 00,2C,65,D4,BC,7C,00,1B,65,CE
998 WG  DATA BA,7C,00,64,62,C8,BC,7C,00,2A
999 Pe  DATA 62,C2,60,00,00,0E,30,39,00,00
1000 37 DATA 30,52,08,00,00,07,66,F4,2C,79
1001 PN DATA 00,00,3A,46,20,79,00,00,3A,56
1002 kk DATA 4E,AE,FF,B8,0C,B9,00,00,00,00
1003 7K DATA 00,00,30,8E,66,00,00,0C,23,FC
1004 er DATA 00,00,00,01,00,00,30,8E,23,F9
1005 XI DATA 00,00,30,8E,00,00,3A,78,2C,79
1006 cq DATA 00,00,3A,3E,20,79,00,00,3A,62
1007 TZ DATA 43,F9,00,00,32,6E,30,3C,00,20
1008 lY DATA 4E,AE,FF,40,23,FC,00,00,00,01
1009 jA DATA 00,00,30,8E,20,3C,00,30,30,31
1010 9k DATA E1,88,23,C0,00,00,30,96,23,C0
1011 Ge DATA 00,00,30,9C,4E,75,08,39,00,05
1012 mU DATA 00,DF,F0,1F,66,00,00,06,60,00
1013 jb DATA 00,AC,48,E7,FF,FE,0C,B9,00,00
1014 Es DATA 00,00,00,00,3A,8A,67,00,00,10
1015 fC DATA 04,B9,00,00,00,01,00,00,3A,8A
1016 K4 DATA 60,00,00,88,20,39,00,00,3A,8E
1017 R3 DATA 90,BC,00,00,00,01,C0,FC,02,80
1018 HN DATA D0,B9,00,00,3A,4A,23,C0,00,00
1019 19 DATA 3B,10,D0,BC,00,00,00,80,23,C0
1020 Iw DATA 00,00,3B,14,D0,BC,00,00,00,80
1021 z2 DATA 23,C0,00,00,3B,18,D0,BC,00,00
1022 9u DATA 00,80,23,C0,00,00,3B,1C,D0,BC
1023 GW DATA 00,00,00,80,23,C0,00,00,3B,20
1024 Yf DATA 33,FC,00,01,00,00,3A,96,23,F9
1025 Ce DATA 00,00,3A,92,00,00,3A,8A,20,39
1026 Ub DATA 00,00,3A,8E,B0,B9,00,00,31,46
1027 g4 DATA 66,00,00,10,23,F9,00,00,30,EA
1028 yJ DATA 00,00,3A,8E,60,00,00,0C,06,B9
1029 Kk DATA 00,00,00,01,00,00,3A,8E,4C,DF
1030 lz DATA 7F,FF,4E,F9,00,00,00,00,2C,79
1031 ca DATA 00,00,3A,3E,42,80,42,81,42,82
1032 oK DATA 42,83,30,39,00,00,3A,72,D0,7C
1033 Qk DATA 00,10,32,10,34,10,36,10,C2,7C
1034 HK DATA 0F,00,E0,59,C4,7C,00,F0,E8,5A
1035 tI DATA C6,7C,00,0F,20,79,00,00,3A,62
1036 3d DATA 4E,AE,FE,E0,4E,75,2C,79,00,00
1037 sL DATA 3A,3E,41,F9,00,00,3A,98,22,79
1038 YN DATA 00,00,3A,66,42,80,42,81,34,3C
1039 9j DATA 01,00,36,3C,00,08,38,3C,00,20
1040 UT DATA 3A,3C,00,20,3C,3C,00,C0,4E,AE
1041 Jq DATA FD,D8,4E,B9,00,00,28,C0,22,79
1042 1R DATA 00,00,3A,66,30,3C,00,02,4E,AE
1043 Wv DATA FE,AA,42,86,22,79,00,00,3A,66
1044 V7 DATA 33,46,00,24,33,7C,00,00,00,26
1045 87 DATA 30,06,32,3C,00,E0,4E,AE,FF,0A
1046 cG DATA 22,79,00,00,3A,66,33,7C,00,00
1047 gW DATA 00,24,33,46,00,26,30,3C,00,E0
1048 9M DATA 32,06,4E,AE,FF,0A,DC,7C,00,07
1049 w8 DATA BC,7C,00,E7,66,C2,22,79,00,00
1050 yH DATA 3A,66,30,3C,00,04,4E,AE,FE,AA
1051 9u DATA 22,79,00,00,3A,66,33,7C,00,70
1052 hz DATA 00,24,33,7C,00,01,00,26,30,3C
1053 Bx DATA 00,70,32,3C,00,DF,4E,AE,FF,0A
1054 pU DATA 22,79,00,00,3A,66,33,7C,00,01
1055 vJ DATA 00,24,33,7C,00,70,00,26,30,3C
1056 vv DATA 00,DF,32,3C,00,70,4E,AE,FF,0A
1057 1B DATA 2C,78,00,04,4E,AE,FF,7C,2C,3C
1058 YC DATA 00,00,00,1F,2E,3C,00,00,00,1F
1059 r3 DATA 20,07,22,06,4E,B9,00,00,2A,94
1060 Qy DATA 20,07,22,06,C0,FC,00,07,C2,FC
1061 sV DATA 00,07,D0,BC,00,00,00,01,D2,BC
1062 oz DATA 00,00,00,01,4E,B9,00,00,29,30
1063 c5 DATA 51,CF,FF,D6,51,CE,FF,CC,2C,78
1064 0A DATA 00,04,4E,AE,FF,76,4E,75,4E,AE
1065 G4 DATA FE,38,2A,7C,00,DF,F0,00,20,79
1066 g5 DATA 00,00,3A,5A,22,28,00,08,4E,B9
1067 Uq DATA 00,00,28,FC,22,28,00,0C,4E,B9
1068 KO DATA 00,00,28,FC,22,28,00,10,4E,B9
1069 bj DATA 00,00,28,FC,22,28,00,14,4E,B9
1070 D6 DATA 00,00,28,FC,4E,AE,FE,32,08,2D
1071 9r DATA 00,0E,00,02,66,F8,2B,41,00,54
1072 LH DATA 3B,7C,00,0C,00,66,3B,7C,FF,FF
1073 ce DATA 00,44,3B,7C,FF,FF,00,46,42,6D
1074 ot DATA 00,42,42,6D,00,74,3B,7C,01,F0
1075 nJ DATA 00,40,3B,7C,38,0E,00,58,4E,75
1076 sT DATA 36,00,C6,BC,00,00,00,0F,24,3C
1077 Ou DATA FC,00,00,00,E6,AA,C2,FC,00,28
1078 aZ DATA E6,48,08,80,00,00,D2,80,76,28
1079 Yc DATA 20,79,00,00,3A,5A,08,05,00,00
1080 Le DATA 67,08,22,68,00,08,D3,C1,61,2C
1081 NE DATA 08,05,00,01,67,08,22,68,00,0C
1082 Tp DATA D3,C1,61,1E,08,05,00,02,67,08
1083 TG DATA 22,68,00,10,D3,C1,61,10,08,05
1084 I9 DATA 00,03,67,08,22,68,00,14,D3,C1
1085 9h DATA 61,02,4E,75,70,05,85,91,D3,C3
1086 uQ DATA 51,C8,FF,FA,4E,75,0C,B9,00,00
1087 8d DATA 00,01,00,00,3A,92,66,00,00,06
1088 M9 DATA 60,00,00,6C,04,B9,00,00,00,01
1089 w2 DATA 00,00,3A,92,41,F9,00,00,3C,3D
1090 e1 DATA 22,39,00,00,3A,92,82,FC,00,64
1091 hZ DATA 61,00,00,C0,82,FC,00,0A,61,00
1092 LK DATA 00,B8,61,00,00,B4,60,00,00,3E
1093 SQ DATA 0C,B9,00,00,00,C8,00,00,3A,92
1094 7d DATA 66,00,00,06,60,00,00,2C,06,B9
1095 iN DATA 00,00,00,01,00,00,3A,92,41,F9
1096 0e DATA 00,00,3C,3D,22,39,00,00,3A,92
1097 1f DATA 82,FC,00,64,61,00,00,80,82,FC
1098 gP DATA 00,0A,61,00,00,78,61,00,00,74
1099 Lk DATA 4B,F9,00;00,3C,3D,2A,3C,00,00
1100 LZ DATA 00,03,2C,3C,00,00,00,63,2E,3C
1101 h0 DATA 00,00,00,5D,4E,B9,00,00,04,18
1102 ai DATA 08,39,00,06,00,BF,E0,01,67,F6
1103 nZ DATA 60,00,F4,BC,41,F9,00,00,3B,74
1104 2p DATA 42,81,32,39,00,00,3A,76,82,FC
1105 ky DATA 00,64,61,00,00,32,82,FC,00,0A
1106 tB DATA 61,00,00,2A,61,00,00,26,4B,F9
1107 Nm DATA 00,00,3B,74,3A,3C,00,03,3C,3C
1108 Qg DATA 01,03,3E,3C,00,C2,23,F9,00,00
1109 Se DATA 3A,66,00,00,3A,6E,4E,B9,00,00
1110 UQ DATA 04,18,4E,75,D2,7C,00,30,10,C1
1111 it DATA 42,41,48,41,4E,75,42,85,20,79
1112 a3 DATA 00,00,3B,1C,4E,B9,00,00,2A,CE
1113 g0 DATA E3,5D,20,79,00,00,3B,18,4E,B9
1114 UF DATA 00,00,2A,CE,E3,5D,20,79,00,00
1115 9Z DATA 3B,14,4E,B9,00,00,2A,CE,E3,5D
1116 zJ DATA 20,79,00,00,3B,10,4E,B9,00,00
1117 Oq DATA 2A,CE,4E,75,26,00,28,01,86,FC
1118 68 DATA 00,08,C8,FC,00,04,D8,43,48,43
1119 ee DATA 14,3C,00,07,94,03,05,30,40,00
1120 DZ DATA 67,00,00,06,08,C5,00,00,4E,75
1121 zh DATA 20,79,00,00,3A,4A,20,39,00,00
1122 80 DATA 3A,78,90,BC,00,00,00,01,D1,FC
1123 OT DATA 00,00,02,00,22,3C,00,00,00,1F
1124 0N DATA 20,FC,FF,FF,FF,FF,51,C9,FF,F8
1125 wS DATA 51,C8,FF,E8,4E,75,20,79,00,00
1126 hV DATA 3A,4A,22,48,24,48,26,48,28,48
```

```
1127 Em  DATA D3,FC,00,00,00,80,D5,FC,00,00
1128 4F  DATA 01,00,D7,FC,00,00,01,80,D9,FC
1129 wm  DATA 00,00,02,00,20,39,00,00,3A,78
1130 aK  DATA 90,BC,00,00,00,01,22,3C,00,00
1131 xm  DATA 00,1F,42,87,24,18,8E,82,24,19
1132 pY  DATA 8E,82,24,1A,8E,82,24,1B,8E,82
1133 4b  DATA 28,C7,51,C9,FF,EA,D1,FC,00,00
1134 XP  DATA 02,00,D3,FC,00,00,02,00,D5,FC
1135 zP  DATA 00,00,02,00,D7,FC,00,00,02,00
1136 Lx  DATA D9,FC,00,00,02,00,51,C8,FF,C2
1137 DW  DATA 4E,75,4E,B9,00,00,01,10,4E,B9
1138 Om  DATA 00,00,00,06,4E,B9,00,00,00,66
1139 Wp  DATA 4E,B9,00,00,2C,9C,4E,B9,00,00
1140 52  DATA 01,54,4E,B9,00,00,24,A0,2C,78
1141 Ha  DATA 00,04,22,3C,00,01,00,03,20,39
1142 to  DATA 00,00,3A,78,C0,FC,02,80,4E,AE
1143 sG  DATA FF,3A,23,C0,00,00,3A,4A,67,DA
1144 aV  DATA 23,C0,00,00,3B,10,D0,BC,00,00
1145 Ag  DATA 00,80,23,C0,00,00,3B,14,D0,BC
1146 pC  DATA 00,00,00,80,23,C0,00,00,3B,18
1147 4z  DATA D0,BC,00,00,00,80,23,C0,00,00
1148 dq  DATA 3B,1C,D0,BC,00,00,00,80,23,C0
1149 AP  DATA 00,00,3B,20,33,FC,00,01,00,00
1150 Ni  DATA 3A,76,4E,B9,00,00,2A,F2,08,39
1151 EJ  DATA 00,06,00,BF,E0,01,67,00,00,10
1152 MM  DATA 08,39,00,0A,00,DF,F0,16,67,00
1153 Gu  DATA 00,BE,66,E6,22,79,00,00,3A,52
1154 RP  DATA 42,85,42,86,3A,29,00,0E,3C,29
1155 D5  DATA 00,0C,BA,7C,00,E0,6C,00,DA,86
1156 9Z  DATA BC,7C,00,E0,6C,00,D8,84,61,00
1157 nq  DATA D9,E0,60,BE,2C,78,00,04,20,39
1158 j7  DATA 00,00,3A,78,C0,FC,02,80,22,79
1159 ez  DATA 00,00,3A,4A,4E,AE,FF,2E,2C,78
1160 J6  DATA 00,04,20,3C,00,00,08,00,22,79
1161 x0  DATA 00,00,3A,5E,4E,AE,FF,2E,4E,B9
1162 G2  DATA 00,00,00,F0,4E,B9,00,00,00,42
1163 pk  DATA 4E,F9,00,00,01,3C,2C,78,00,04
1164 Nt  DATA 20,3C,00,00,08,00,22,3C,00,01
1165 RK  DATA 00,03,4E,AE,FF,3A,23,C0,00,00
1166 QH  DATA 3A,5E,22,40,20,7C,00,00,36,3E
1167 3N  DATA 20,3C,00,00,01,FF,22,D8,51,C8
1168 5p  DATA FF,FC,22,39,00,00,3A,5E,41,F9
1169 7f  DATA 00,00,34,36,D1,FC,00,00,00,0A
1170 kg  DATA 20,3C,00,00,00,19,D3,90,D1,FC
1171 nF  DATA 00,00,00,14,51,C8,FF,F6,4E,75
1172 aL  DATA 22,79,00,00,3A,52,3A,29,00,0E
1173 zM  DATA 3C,29,00,0C,BA,7C,00,E0,64,00
1174 gN  DATA 00,2E,BC,7C,00,E0,64,00,00,26
1175 5t  DATA 33,F9,00,00,3A,72,00,00,3A,80
1176 n0  DATA 33,FC,00,00,00,00,3A,72,61,00
1177 x5  DATA D9,18,33,F9,00,00,3A,80,00,00
1178 LN  DATA 3A,72,60,00,FE,EC,BA,7C,01,3F
1179 hH  DATA 66,00,FE,E4,BC,7C,00,00,66,00
1180 Yh  DATA FE,DC,60,00,FF,1A,00,00,00,00
1181 xR  DATA 01,40,01,00,00,05,00,01,00,02
1182 C2  DATA 00,0F,00,00,00,00,00,00,00,00
1183 5L  DATA 00,00,00,00,00,00,00,00,00,00
1184 hk  DATA 00,00,01,40,01,00,00,01,00,00
1185 7f  DATA 00,00,00,01,18,80,00,00,00,00
1186 80  DATA 00,00,00,00,00,00,00,00,00,00
1187 Tk  DATA 00,00,00,00,00,00,01,40,01,00
1188 oY  DATA 01,40,01,00,00,0F,00,01,00,01
1189 sB  DATA 00,DF,00,DF,02,03,00,00,00,00
1190 rC  DATA 00,01,18,00,00,00,00,00,00,00
1191 DT  DATA 00,00,00,00,00,00,00,00,00,00
1192 Og  DATA 00,00,00,00,00,DF,00,DF,00,DF
1193 Q1  DATA 00,DF,00,0F,00,58,00,5E,00,90
1194 f1  DATA 00,2D,02,03,00,00,00,00,00,01
1195 Jv  DATA 18,00,00,00,30,46,00,00,00,00
1196 IY  DATA 00,00,00,00,00,00,00,00,00,00
1197 Ov  DATA 00,00,00,90,00,2D,00,90,00,2D
1198 yg  DATA 00,0F,00,00,2E,76,00,60,00,57
1199 DL  DATA 00,20,00,0A,00,00,08,01,00,04
1200 4b  DATA 00,00,2E,22,00,00,00,00,00,00
1201 zc  DATA 00,00,00,00,00,00,00,00,2E,46
1202 Ug  DATA 00,01,00,00,00,00,00,00,00,00
1203 BK  DATA 03,00,00,05,00,00,2E,32,00,00
1204 gd  DATA 00,00,FF,FE,FF,FE,00,20,FF,FE
1205 Lx  DATA 00,20,00,09,FF,FE,00,09,FF,FE
1206 CE  DATA FF,FE,00,00,2E,6A,00,00,2E,70
1207 bv  DATA 00,00,00,04,00,00,00,00,00,00
1208 Uk  DATA 00,00,00,00,00,00,00,00,00,00
1209 z6  DATA 00,00,00,00,00,00,00,00,30,30
1210 Ih  DATA 31,00,00,00,30,30,31,00,00,00
1211 Bv  DATA 00,00,00,00,00,27,00,75,00,90
1212 Vk  DATA 00,0A,00,00,00,01,00,04,00,00
1213 Dz  DATA 2E,A2,00,00,00,00,00,00,00,00
1214 Fc  DATA 00,00,00,00,00,00,2E,C6,00,02
1215 h0  DATA 00,00,00,00,00,00,00,00,03,00
1216 bE  DATA 00,05,00,00,2E,B2,00,00,00,00
1217 hD  DATA FF,FE,FF,FE,00,90,FF,FE,00,90
1218 WE  DATA 00,09,FF,FE,00,09,FF,FE,FF,FE
1219 IX  DATA 00,00,2E,EA,00,00,2F,14,00,00
1220 c0  DATA 00,28,00,00,00,00,00,00,00,00
1221 hx  DATA 00,00,00,00,00,00,00,00,00,00
1222 Vb  DATA 00,00,00,00,00,00,4F,62,6A,65
1223 Zh  DATA 6B,74,65,2E,41,4F,45,00,00,00
1224 k0  DATA 00,00,00,00,00,00,00,00,00,00
1225 11  DATA 00,00,00,00,00,00,00,00,00,00
1226 Qu  DATA 00,00,00,00,00,00,00,00,4F,62
1227 Ug  DATA 6A,65,6B,74,65,2E,41,4F,45,00
1228 o4  DATA 00,00,00,00,00,00,00,00,00,00
1229 p5  DATA 00,00,00,00,00,00,00,00,00,00
1230 q6  DATA 00,00,00,00,00,00,00,00,00,00
1231 6q  DATA 00,00,2F,9A,00,38,00,57,00,20
1232 Tq  DATA 00,0A,00,00,08,01,00,04,00,00
1233 Kr  DATA 2E,22,00,00,00,00,00,00,00,00
1234 fv  DATA 00,00,00,00,00,00,2F,6A,00,01
1235 xI  DATA 00,00,00,00,00,00,2F,8E,00,00
1236 Ny  DATA 2F,94,00,00,00,04,00,00,00,00
1237 xD  DATA 00,00,00,00,00,00,00,00,00,00
1238 yE  DATA 00,00,00,00,00,00,00,00,00,00
1239 Fa  DATA 30,30,31,00,00,00,30,30,31,00
1240 hF  DATA 00,00,00,00,2F,F6,00,88,00,57
1241 t1  DATA 00,20,00,0A,00,00,08,01,00,04
1242 kH  DATA 00,00,2E,22,00,00,00,00,00,00
1243 fY  DATA 00,00,00,00,00,00,00,00,2F,C6
1244 5v  DATA 00,04,00,00,00,00,00,00,2F,EA
1245 zH  DATA 00,00,2F,F0,00,00,00,04,00,00
1246 6M  DATA 00,00,00,00,00,00,00,00,00,00
1247 7N  DATA 00,00,00,00,00,00,00,00,00,00
1248 KM  DATA 00,00,30,30,31,00,00,00,30,30
1249 Qm  DATA 31,00,00,00,00,00,00,00,00,20
1250 zk  DATA 00,75,00,90,00,0A,00,00,00,01
1251 2i  DATA 00,04,00,00,2E,A2,00,00,00,00
1252 CS  DATA 00,00,00,00,00,00,00,00,00,00
1253 p7  DATA 30,22,00,05,00,00,00,00,00,00
1254 Y3  DATA 2E,EA,00,00,2F,14,00,00,00,28
1255 FV  DATA 00,00,00,00,00,00,00,00,00,00
1256 GW  DATA 00,00,00,00,00,00,00,00,00,00
1257 Ce  DATA 00,00,00,00,00,00,00,00,00,39
1258 mu  DATA 00,0F,00,20,00,0A,00,00,08,01
1259 g7  DATA 00,04,00,00,2E,22,00,00,00,00
1260 Ka  DATA 00,00,00,00,00,00,00,00,00,00
1261 EX  DATA 30,72,00,01,00,00,00,00,00,00
1262 U7  DATA 30,96,00,00,30,9C,00,00,00,04
1263 Nd  DATA 00,00,00,00,00,00,00,00,00,00
1264 6W  DATA 00,00,00,00,00,00,00,00,00,64
1265 1R  DATA 00,00,00,00,31,30,30,00,00,00
1266 MH  DATA 31,30,30,00,00,00,00,00,30,FE
1267 Ex  DATA 00,38,00,61,00,20,00,0A,00,00
1268 Uc  DATA 08,01,00,04,00,00,2E,22,00,00
1269 TJ  DATA 00,00,00,00,00,00,00,00,00,00
1270 Cd  DATA 00,00,30,CE,00,01,00,00,00,00
1271 f7  DATA 00,00,30,F2,00,00,30,F8,00,00
1272 uu  DATA 00,04,00,00,00,00,00,00,00,00
1273 Xn  DATA 00,00,00,00,00,00,00,00,00,00
1274 MT  DATA 00,01,00,00,00,00,30,30,31,00
1275 HS  DATA 00,00,30,30,31,00,00,00,00,00
1276 Kc  DATA 00,00,00,88,00,61,00,20,00,0A
1277 VF  DATA 00,00,08,01,00,04,00,00,2E,22
1278 cs  DATA 00,00,00,00,00,00,00,00,00,00
1279 3k  DATA 00,00,00,00,31,2A,00,04,00,00
1280 2J  DATA 00,00,00,00,31,4E,00,00,31,54
1281 n7  DATA 00,00,00,04,00,00,00,00,00,00
1282 gw  DATA 00,00,00,00,00,00,00,00,00,00
1283 FO  DATA 00,00,00,02,00,00,00,00,30,30
1284 dz  DATA 32,00,00,00,30,30,32,00,00,00
1285 Bq  DATA 00,00,31,B6,00,38,00,57,00,20
1286 L1  DATA 00,0A,00,00,08,01,00,04,00,00
1287 Cj  DATA 2E,22,00,00,00,00,00,00,00,00
1288 yS  DATA 00,00,00,00,00,00,31,86,00,01
1289 3q  DATA 00,00,00,00,00,00,31,AA,00,00
1290 bH  DATA 31,B0,00,00,00,04,00,00,00,00
1291 p5  DATA 00,00,00,00,00,00,00,00,00,00
1292 tA  DATA 00,00,00,00,00,01,00,00,00,00
1293 7S  DATA 30,30,31,00,00,00,30,30,31,00
1294 9z  DATA 00,00,00,00,32,12,00,88,00,57
1295 1t  DATA 00,20,00,0A,00,00,08,01,00,04
1296 c9  DATA 00,00,2E,22,00,00,00,00,00,00
1297 1D  DATA 00,00,00,00,00,00,00,00,31,E2
1298 A6  DATA 00,04,00,00,00,00,00,00,32,06
1299 6o  DATA 00,00,32,0C,00,00,00,04,00,00
1300 yE  DATA 00,00,00,00,00,00,00,00,00,00
1301 7P  DATA 00,00,00,00,00,00,00,02,00,00
1302 HK  DATA 00,00,30,30,32,00,00,00,30,30
1303 Cx  DATA 32,00,00,00,00,00,00,00,00,7C
1304 c6  DATA 00,6B,00,20,00,0A,00,00,08,01
1305 Qr  DATA 00,04,00,00,2E,22,00,00,00,00
1306 4K  DATA 00,00,00,00,00,00,00,00,00,00
1307 gm  DATA 32,3E,00,04,00,00,00,00,00,00
1308 Eu  DATA 32,62,00,00,32,68,00,00,00,04
1309 7N  DATA 00,00,00,00,00,00,00,00,00,00
1310 DU  DATA 00,00,00,00,00,00,00,00,00,01
1311 m7  DATA 00,00,00,00,30,30,31,00,00,00
1312 bG  DATA 30,30,31,00,00,00,00,00,05,55
1313 5U  DATA 09,99,0B,BB,0E,EE,00,00,00,00
1314 CS  DATA 00,00,00,00,00,00,00,00,00,00
1315 DT  DATA 00,00,00,00,00,00,00,00,00,00
1316 5C  DATA 00,07,00,0F,07,7F,07,07,0F,0F
1317 Ab  DATA 0F,7F,03,33,0A,AA,0F,FF,07,00
1318 9p  DATA 0F,00,0F,77,00,70,00,F0,07,F7
1319 K2  DATA 00,02,00,E8,00,0C,00,F8,00,10
1320 uV  DATA 00,E8,00,1A,00,F8,00,1E,00,E8
1321 pG  DATA 00,28,00,F8,00,2C,00,E8,00,36
1322 IY  DATA 00,F8,00,3A,00,E8,00,44,00,F8
1323 Nj  DATA 00,48,00,E8,00,52,00,F8,00,56
1324 3r  DATA 00,E8,00,60,00,F8,00,64,00,E8
1325 7r  DATA 00,6E,00,F8,00,72,00,E8,00,7C
1326 KG  DATA 00,F8,00,80,00,E8,00,8A,00,F8
1327 k1  DATA 00,8E,00,E8,00,98,00,F8,00,9C
1328 kK  DATA 00,E8,00,A6,00,F8,00,AA,00,E8
1329 Ym  DATA 00,B4,00,F8,00,B8,00,E8,00,C2
1330 8N  DATA 00,F8,00,C6,00,E8,00,D0,00,F8
1331 1A  DATA 00,D4,00,E8,00,DE,00,F8,01,09
1332 2L  DATA 00,50,01,18,00,5F,01,09,00,2E
1333 jX  DATA 01,18,00,3D,01,1A,00,3F,01,29
1334 vS  DATA 00,4E,00,F8,00,3F,01,07,00,4E
1335 Xy  DATA 01,09,00,3F,01,18,00,4E,01,00
1336 wd  DATA 00,66,01,0F,00,75,01,11,00,66
1337 Iq  DATA 01,20,00,75,01,00,00,77,01,0F
1338 19  DATA 00,86,01,11,00,77,01,20,00,86
1339 tS  DATA 01,00,00,88,01,0F,00,97,01,11
1340 KZ  DATA 00,88,01,20,00,97,01,00,00,99
1341 3K  DATA 01,0F,00,A8,01,11,00,99,01,20
1342 d1  DATA 00,A8,00,F8,00,BC,00,FF,00,C3
1343 hJ  DATA 01,20,00,BC,01,27,00,C3,00,FB
1344 w2  DATA 00,E2,01,02,00,E9,00,FB,00,F6
1345 4r  DATA 01,02,00,FD,01,0B,00,E2,01,12
1346 Ef  DATA 00,E9,01,0B,00,F6,01,12,00,FD
1347 5m  DATA 01,1B,00,E2,01,22,00,E9,01,1B
1348 xK  DATA 00,F6,01,22,00,FD,01,01,00,BC
1349 Oq  DATA 01,25,00,C3,00,00,07,42,00,00
1350 kS  DATA 07,A0,00,00,07,FA,00,00,08,48
1351 db  DATA 00,00,08,96,00,00,0A,28,00,00
1352 mT  DATA 14,6E,00,00,14,A4,00,00,15,38
1353 tS  DATA 00,00,15,84,00,00,16,02,00,00
1354 x0  DATA 19,A0,00,00,1A,F4,00,00,20,46
1355 v7  DATA 00,00,20,A2,00,00,21,00,00,00
1356 bG  DATA 21,3C,00,00,21,78,00,00,21,B4
1357 96  DATA 00,00,21,F0,00,00,22,28,00,00
1358 f7  DATA 22,60,00,00,00,00,00,10,00,10
1359 PL  DATA 00,02,00,00,00,00,03,00,00,00
1360 Au  DATA 34,4A,00,00,FF,DE,00,10,00,10
1361 ZZ  DATA 00,02,00,00,00,40,03,00,00,00
1362 sM  DATA 34,5E,00,11,FF,EF,00,10,00,10
1363 jn  DATA 00,02,00,00,00,80,03,00,00,00
1364 Kp  DATA 34,72,FF,EF,FF,EF,00,10,00,10
1365 7M  DATA 00,02,00,00,00,C0,03,00,00,00
1366 b9  DATA 34,86,00,00,FF,EF,00,10,00,10
1367 cZ  DATA 00,02,00,00,01,00,03,00,00,00
1368 BT  DATA 00,00,00,00,00,00,00,10,00,10
1369 hh  DATA 00,02,00,00,00,40,03,00,00,00
1370 1S  DATA 34,AE,00,00,FF,DE,00,10,00,10
1371 bX  DATA 00,02,00,00,00,00,03,00,00,00
1372 Q7  DATA 34,C2,FF,EF,FF,EF,00,10,00,10
1373 tx  DATA 00,02,00,00,00,80,03,00,00,00
1374 pW  DATA 34,D6,00,11,FF,EF,00,10,00,10
1375 HW  DATA 00,02,00,00,00,C0,03,00,00,00
1376 sv  DATA 34,EA,00,00,FF,EF,00,10,00,10
```

Listing 1. (Fortsetzung)

```
1377 uv  DATA 00,02,00,00,01,40,03,00,00,00
1378 Ld  DATA 00,00,00,00,00,00,00,10,00,10
1379 49  DATA 00,02,00,00,01,80,03,00,00,00
1380 7N  DATA 35,12,00,11,00,00,00,10,00,10
1381 Si  DATA 00,02,00,00,01,C0,03,00,00,00
1382 Zf  DATA 35,26,00,00,00,11,00,10,00,10
1383 xv  DATA 00,02,00,00,02,00,03,00,00,00
1384 sE  DATA 35,3A,00,11,00,11,00,10,00,10
1385 79  DATA 00,02,00,00,02,40,03,00,00,00
1386 KW  DATA 35,4E,00,00,00,22,00,10,00,10
1387 HN  DATA 00,02,00,00,02,80,03,00,00,00
1388 oD  DATA 35,62,00,11,00,22,00,10,00,10
1389 fw  DATA 00,02,00,00,02,C0,03,00,00,00
1390 GV  DATA 35,76,00,00,00,33,00,10,00,10
1391 A9  DATA 00,02,00,00,03,00,03,00,00,00
1392 Z4  DATA 35,8A,00,11,00,33,00,10,00,10
1393 KN  DATA 00,02,00,00,03,40,03,00,00,00
1394 eA  DATA 00,00,00,00,00,00,00,08,00,08
1395 Ub  DATA 00,02,00,00,03,80,03,00,00,00
1396 7v  DATA 35,B2,00,28,00,00,00,08,00,08
1397 o4  DATA 00,02,00,00,03,A0,03,00,00,00
1398 iE  DATA 00,00,00,00,00,00,00,08,00,08
1399 uC  DATA 00,02,00,00,03,C0,03,00,00,00
1400 ZW  DATA 35,DA,00,00,00,14,00,08,00,08
1401 OK  DATA 00,02,00,00,03,E0,03,00,00,00
1402 wV  DATA 35,EE,00,10,00,00,00,08,00,08
1403 yG  DATA 00,02,00,00,03,C0,03,00,00,00
1404 iG  DATA 36,02,00,10,00,14,00,08,00,08
1405 40  DATA 00,02,00,00,03,E0,03,00,00,00
1406 5F  DATA 36,16,00,20,00,00,00,08,00,08
1407 2K  DATA 00,02,00,00,03,C0,03,00,00,00
1408 U2  DATA 36,2A,00,20,00,14,00,08,00,08
1409 8S  DATA 00,02,00,00,03,E0,03,00,00,00
1410 TO  DATA 00,00,00,00,7B,FE,7B,FE,7B,FE
1411 hx  DATA 7B,FE,7B,FE,7B,FE,7B,FE,7B,FE
1412 EQ  DATA 5F,FE,6F,FE,77,FE,7B,FE,7D,FE
1413 G5  DATA 7E,FE,00,00,00,01,00,21,00,21
1414 m2  DATA 00,21,00,21,00,21,00,21,00,21
1415 OX  DATA 00,21,00,05,00,09,00,11,00,21
1416 bt  DATA 00,41,00,81,7F,FF,00,00,7F,7E
1417 FF  DATA 7F,BE,7F,DE,7F,EE,7F,F6,7F,FA
1418 Si  DATA 7F,DE,7F,DE,7F,DE,7F,DE,7F,DE
1419 AW  DATA 7F,DE,7F,DE,7F,DE,00,00,00,01
1420 b0  DATA 01,01,02,01,04,01,08,01,10,01
1421 Pd  DATA 20,01,04,01,04,01,04,01,04,01
1422 vi  DATA 04,01,04,01,04,01,04,01,7F,FF
1423 5s  DATA 00,00,7F,FE,7F,FE,7F,FE,7F,FE
1424 ku  DATA 7F,FE,7F,FE,7F,FE,7F,FC,7F,FA
1425 u0  DATA 00,76,7F,EE,7F,DE,7F,BE,7F,FE
1426 Yy  DATA 00,00,00,01,00,01,00,41,00,21
1427 Xz  DATA 00,11,7F,89,00,05,00,03,00,01
1428 Mc  DATA 00,01,00,01,00,01,00,01,00,01
1429 Iq  DATA 00,01,7F,FF,00,00,7F,FE,7D,FE
1430 iA  DATA 7B,FE,77,FE,6E,00,5F,FE,3F,FE
1431 JZ  DATA 7F,FE,7F,FE,7F,FE,7F,FE,7F,FE
1432 u3  DATA 7F,FE,7F,FE,00,00,00,01,00,01
1433 Rh  DATA 00,01,00,01,00,01,00,01,00,01
1434 dl  DATA 00,01,40,01,20,01,11,FF,08,01
1435 pY  DATA 04,01,02,01,00,01,7F,FF,00,00
1436 MX  DATA 7F,BE,7F,BE,7F,DE,7F,BE,6F,7E
1437 1L  DATA 17,FE,FB,FE,7F,DE,7F,E8,7E,F6
1438 Ds  DATA 7D,FE,7B,FE,7D,FE,7D,FE,00,00
1439 HN  DATA 00,01,02,01,02,01,04,01,02,01
1440 xG  DATA 01,09,00,17,00,21,04,01,68,01
1441 9S  DATA 10,81,00,41,00,21,00,41,00,41
1442 Dg  DATA 7F,FF,00,00,7F,7E,7F,BE,7F,DE
1443 bb  DATA 7F,BE,6F,BE,53,FE,3F,FE,7F,FC
1444 f3  DATA 7F,CA,7D,F6,7D,FE,7B,FE,7D,FE
1445 iR  DATA 7E,FE,00,00,00,01,01,01,02,01
1446 5k  DATA 04,01,02,01,02,09,00,35,00,03
1447 dT  DATA 40,01,2C,01,10,41,00,41,00,21
1448 5c  DATA 00,41,00,81,7F,FF,00,00,7F,FE
1449 hD  DATA 47,CE,5F,F6,5F,FA,5F,FA,5F,FA
1450 gw  DATA 5F,FA,5F,FA,5F,FA,5F,FA,5F,FA
1451 LZ  DATA 5F,FA,40,02,7F,FE,00,00,00,01
1452 JJ  DATA 00,01,07,F1,07,49,07,45,07,C5
1453 3J  DATA 00,05,00,05,00,05,00,05,00,05
1454 Gx  DATA 00,05,00,05,1F,FD,00,01,7F,FF
1455 wb  DATA 00,00,7F,FE,03,86,3F,BE,3B,B6
1456 7q  DATA 3B,B6,3B,B6,3B,B6,3B,BE,3B,AE
1457 Qo  DATA 3B,AE,22,36,7F,F6,7F,FE,7F,FE
1458 YC  DATA 00,00,00,01,00,01,00,01,3E,39
1459 To  DATA 22,25,22,25,22,25,22,25,22,39
1460 qw  DATA 22,29,22,29,22,25,3F,E5,00,01
1461 kN  DATA 00,01,7F,FF,00,00,7E,FE,7E,FE
1462 Cj  DATA 7E,FE,76,EE,68,96,5E,FA,3E,FC
1463 V1  DATA 7E,FE,7E,FE,7E,FE,7E,FE,7E,FE
1464 JQ  DATA 7E,FE,7E,FE,00,00,00,01,00,01
1465 Xn  DATA 00,81,00,81,00,81,00,81,00,81
1466 Gp  DATA 00,81,40,83,20,85,16,E9,08,91
1467 h4  DATA 00,81,00,81,00,81,7F,FF,00,00
1468 ps  DATA 7E,FE,7D,FE,7B,FE,77,FE,7B,FE
1469 ik  DATA 7B,FE,00,02,7F,FE,7B,FE,7B,FE
1470 Ca  DATA 77,FE,7B,FE,7D,FE,7E,FE,00,00
1471 7y  DATA 00,01,00,81,00,41,00,21,00,11
1472 Lr  DATA 00,21,00,21,00,01,3F,FF,00,21
1473 7Z  DATA 00,21,00,11,00,21,00,41,00,81
1474 zp  DATA 7F,FF,00,00,7F,FE,77,FE,6F,FE
1475 fu  DATA 5F,FE,6F,FE,6F,FE,6F,FE,6F,FE
1476 3I  DATA 6F,FE,6F,FE,6F,FE,6F,FE,60,02
1477 F5  DATA 7F,FE,00,00,00,01,00,01,04,01
1478 kz  DATA 02,01,01,01,02,01,02,01,02,01
1479 r1  DATA 02,01,02,01,03,FD,00,05,00,05
1480 VU  DATA 00,01,00,01,7F,FF,00,00,7F,FE
1481 Ox  DATA 40,0E,5F,EE,5F,FE,5F,EE,5F,EE
1482 G1  DATA 5F,EE,5F,EE,5F,EE,5F,EE,48,0E
1483 cu  DATA 7F,FE,7F,FE,7F,FE,00,00,00,01
1484 Gx  DATA 00,01,00,01,00,01,0F,FD,08,05
1485 9P  DATA 08,05,08,05,08,05,08,05,08,05
1486 cY  DATA 08,05,08,05,0F,FD,00,01,7F,FF
1487 U0  DATA 00,00,7F,FE,7C,FE,79,FE,7B,FE
1488 40  DATA 7F,FE,78,3E,7D,FE,7D,FE,7D,FE
1489 G5  DATA 7D,FE,7D,FE,71,DE,7F,FE,7F,FE
1490 WA  DATA 00,00,00,01,00,01,00,01,01,C1
1491 aS  DATA 03,E1,01,C1,00,01,03,C1,01,C1
1492 Ed  DATA 01,C1,01,C1,01,C1,01,C1,07,F1
1493 g1  DATA 00,01,7F,FF,00,00,7F,FE,63,FE
1494 FH  DATA 5D,FE,5D,FE,5D,FE,63,FE,77,FE
1495 aS  DATA 78,3E,77,FE,47,FE,47,FE,77,FE
1496 Fu  DATA 77,FE,77,FE,00,00,00,01,00,01
1497 Bg  DATA 00,01,00,31,00,49,00,49,00,31
1498 iC  DATA 00,21,00,11,00,09,00,05,00,05
1499 7U  DATA 00,05,07,FD,00,01,7F,FF,00,00
1500 cS  DATA 6E,00,52,00,3E,00,7E,00,7E,00
1501 fL  DATA 7E,00,00,00,01,00,01,00,01,00
1502 qP  DATA 01,00,41,00,2D,00,11,00,7F,00
1503 yk  DATA 00,00,7E,00,7E,00,7E,00,7C,00
1504 KD  DATA 4A,00,76,00,00,00,01,00,09,00
1505 1Q  DATA 35,00,03,00,01,00,01,00,01,00
1506 1p  DATA 7F,00,00,00,76,00,7A,00,7C,00
1507 yT  DATA 7A,00,7A,00,7E,00,00,00,01,00
1508 Of  DATA 11,00,21,00,41,00,21,00,21,00
1509 uQ  DATA 01,00,7F,00,00,00,7E,00,5E,00
1510 Zn  DATA 5E,00,3E,00,5E,00,6E,00,00,00
1511 TZ  DATA 01,00,01,00,05,00,05,00,03,00
1512 RO  DATA 05,00,09,00,7F,00,00,00,00,00
1513 Pf  DATA 00,00,00,00,00,00,00,00,00,00
1514 Qg  DATA 00,00,00,00,00,00,00,00,00,00
1515 Rh  DATA 00,00,00,00,00,00,00,00,00,00
1516 Si  DATA 00,00,00,00,00,00,00,00,00,00
1517 Yp  DATA 00,00,00,00,00,00,00,00,00,01
1518 Wn  DATA 00,00,00,01,00,00,00,00,00,00
1519 V1  DATA 00,00,00,00,00,00,00,00,00,00
1520 Wm  DATA 00,00,00,00,00,00,00,00,00,00
1521 h2  DATA 00,00,00,05,00,00,00,00,00,00
1522 Yo  DATA 00,00,00,00,00,00,00,00,00,00
1523 Zp  DATA 00,00,00,00,00,00,00,00,00,00
1524 aq  DATA 00,00,00,00,00,00,00,00,00,00
1525 br  DATA 00,00,00,00,00,00,00,00,00,00
1526 cs  DATA 00,00,00,00,00,00,00,00,00,00
1527 dt  DATA 00,00,00,00,00,00,00,00,00,00
1528 eu  DATA 00,00,00,00,00,00,00,00,00,00
1529 fv  DATA 00,00,00,00,00,00,00,00,00,00
1530 gw  DATA 00,00,00,00,00,00,00,00,00,00
1531 hx  DATA 00,00,00,00,00,00,00,00,00,00
1532 2M  DATA 00,00,00,00,00,00,00,00,00,04
1533 3Q  DATA 00,20,00,05,00,00,00,00,00,00
1534 k0  DATA 00,00,00,00,00,00,00,00,00,00
1535 11  DATA 00,00,00,00,00,00,00,00,00,00
1536 CK  DATA 00,00,00,00,00,00,00,00,67,72
1537 P9  DATA 61,70,68,69,63,73,2E,6C,69,62
1538 TM  DATA 72,61,72,79,00,64,6F,73,2E,6C
1539 OP  DATA 69,62,72,61,72,79,00,69,6E,74
1540 dO  DATA 75,69,74,69,6F,6E,2E,6C,69,62
1541 29  DATA 72,61,72,79,00,4F,62,6A,65,6B
1542 FM  DATA 74,20,4E,72,2E,52,20,47,20,42
1543 Fs  DATA 30,20,30,20,30,20,30,30,31,4C
1544 Ei  DATA 61,64,65,6E,53,70,65,69,63,68
1545 VC  DATA 65,72,6E,4C,61,64,65,6E,20,6E
1546 x8  DATA 61,63,68,20,4F,62,6A,65,6B,74
1547 AR  DATA 44,61,74,65,69,6E,61,6D,65,46
1548 ZS  DATA 65,72,74,69,67,53,70,65,69,63
1549 OO  DATA 68,65,72,6E,20,76,6F,6E,20,4F
1550 QV  DATA 62,6A,65,6B,74,6E,61,63,68,41
1551 YH  DATA 4D,49,47,41,20,4F,42,4A,45,43
1552 oM  DATA 54,45,44,49,54,4F,52,20,56,31
1553 aN  DATA 2E,32,77,72,69,74,74,65,6E,20
1554 32  DATA 31,39,38,38,20,62,79,4D,49,43
1555 Fk  DATA 48,41,45,4C,20,42,45,52,54,53
1556 K3  DATA 43,48,A9,20,4D,61,72,6B,74,20
1557 KC  DATA 26,20,54,65,63,68,6E,69,6B,5A
1558 N6  DATA 65,69,67,65,20,4F,62,6A,65,6B
1559 iU  DATA 74,20,4E,72,2E,41,62,62,72,75
1560 1r  DATA 63,68,41,6E,69,6D,61,74,69,6F
1561 V1  DATA 6E,20,76,6F,6E,20,4F,62,6A,65
1562 dJ  DATA 6B,74,47,65,73,63,68,77,69,6E
1563 GZ  DATA 64,69,67,6B,65,69,74,30,30,35
1564 bb  DATA 4D,61,78,2E,20,4F,62,6A,65,6B
1565 bO  DATA 74,7A,61,68,6C,4B,6F,70,69,65
1566 qz  DATA 72,65,6E,20,76,6F,6E,20,4F,62
1567 Oa  DATA 6A,65,6B,74,41,6E,7A,61,68,6C
1568 3I  DATA 00,00,03,EC,00,00,03,56,00,00
1569 wX  DATA 00,00,00,00,00,02,00,00,00,0C
1570 51  DATA 00,00,00,18,00,00,00,1E,00,00
1571 yZ  DATA 00,2A,00,00,00,30,00,00,00,3C
1572 Nc  DATA 00,00,00,48,00,00,00,52,00,00
1573 oI  DATA 00,5C,00,00,00,68,00,00,00,72
1574 CT  DATA 00,00,00,7C,00,00,00,82,00,00
1575 Kw  DATA 00,88,00,00,00,8E,00,00,00,98
1576 7t  DATA 00,00,00,A4,00,00,00,AA,00,00
1577 Ca  DATA 00,B2,00,00,00,BA,00,00,00,C6
1578 Oh  DATA 00,00,00,CC,00,00,00,D2,00,00
1579 PI  DATA 00,D8,00,00,00,E6,00,00,00,F2
1580 62  DATA 00,00,00,F8,00,00,01,02,00,00
1581 V3  DATA 01,36,00,00,01,3E,00,00,01,48
1582 pC  DATA 00,00,01,56,00,00,01,5C,00,00
1583 Ej  DATA 01,66,00,00,01,6A,00,00,01,70
1584 14  DATA 00,00,01,7E,00,00,01,94,00,00
1585 iP  DATA 01,A2,00,00,01,AE,00,00,01,BE
1586 Ew  DATA 00,00,01,D8,00,00,01,FC,00,00
1587 1L  DATA 02,0A,00,00,02,28,00,00,02,46
1588 4N  DATA 00,00,02,54,00,00,02,6C,00,00
1589 JO  DATA 02,86,00,00,02,94,00,00,02,B6
1590 GY  DATA 00,00,02,D0,00,00,02,D6,00,00
1591 Cc  DATA 02,E4,00,00,02,FE,00,00,03,0A
1592 Ko  DATA 00,00,03,1C,00,00,03,44,00,00
1593 1Q  DATA 03,5E,00,00,03,64,00,00,03,6A
1594 OD  DATA 00,00,03,70,00,00,03,82,00,00
1595 qY  DATA 03,88,00,00,03,9A,00,00,03,A0
1596 Qp  DATA 00,00,03,B2,00,00,03,B8,00,00
1597 V7  DATA 03,CA,00,00,03,DC,00,00,03,E2
1598 g4  DATA 00,00,03,F4,00,00,03,FA,00,00
1599 qP  DATA 04,0C,00,00,04,12,00,00,04,1A
1600 BG  DATA 00,00,04,20,00,00,04,36,00,00
1601 4w  DATA 04,42,00,00,04,50,00,00,04,6C
1602 CU  DATA 00,00,04,7A,00,00,04,8A,00,00
1603 Vo  DATA 04,90,00,00,04,9A,00,00,04,B0
1604 5w  DATA 00,00,04,BE,00,00,04,CC,00,00
1605 mM  DATA 04,DE,00,00,05,04,00,00,05,22
1606 Qk  DATA 00,00,05,26,00,00,05,2C,00,00
1607 p1  DATA 05,32,00,00,05,3E,00,00,05,62
1608 B7  DATA 00,00,05,68,00,00,05,74,00,00
1609 y5  DATA 05,C6,00,00,05,D2,00,00,05,E0
1610 mR  DATA 00,00,05,EE,00,00,05,FC,00,00
1611 5j  DATA 06,02,00,00,06,14,00,00,06,18
1612 71  DATA 00,00,06,1E,00,00,06,3C,00,00
1613 r5  DATA 06,7A,00,00,06,84,00,00,06,8E
1614 RR  DATA 00,00,06,9C,00,00,06,A2,00,00
1615 Ov  DATA 06,B0,00,00,06,BE,00,00,06,D8
1616 A3  DATA 00,00,06,FE,00,00,07,22,00,00
1617 hR  DATA 07,36,00,00,07,3A,00,00,07,46
1618 30  DATA 00,00,07,50,00,00,07,5A,00,00
1619 kx  DATA 07,64,00,00,07,6E,00,00,07,A4
1620 yY  DATA 00,00,07,AE,00,00,07,B8,00,00
1621 ej  DATA 07,C2,00,00,07,CC,00,00,07,FE
1622 tw  DATA 00,00,08,08,00,00,08,12,00,00
1623 XT  DATA 08,1C,00,00,08,26,00,00,08,4C
1624 SC  DATA 00,00,08,56,00,00,08,60,00,00
1625 OF  DATA 08,6A,00,00,08,74,00,00,08,9A
1626 Db  DATA 00,00,08,A4,00,00,08,AA,00,00
1627 RA  DATA 08,B0,00,00,08,C8,00,00,08,D2
1628 Vp  DATA 00,00,08,DC,00,00,08,E6,00,00
```

```
1629 X2  DATA 08,F0,00,00,09,24,00,00,09,2E
1630 cT  DATA 00,00,09,38,00,00,09,42,00,00
1631 7N  DATA 09,74,00,00,09,7E,00,00,09,88
1632 yd  DATA 00,00,09,92,00,00,09,B6,00,00
1633 NL  DATA 09,C0,00,00,09,CA,00,00,09,D4
1634 Sg  DATA 00,00,09,FC,00,00,0A,02,00,00
1635 SM  DATA 0A,08,00,00,0A,20,00,00,0A,2A
1636 fE  DATA 00,00,0A,30,00,00,0A,3A,00,00
1637 5z  DATA 0A,48,00,00,0A,56,00,00,0A,60
1638 Ky  DATA 00,00,0A,6E,00,00,0A,88,00,00
1639 Ty  DATA 0A,8C,00,00,0A,92,00,00,0A,A4
1640 Uh  DATA 00,00,0A,AA,00,00,0A,BC,00,00
1641 p9  DATA 0A,C2,00,00,0A,D4,00,00,0A,EE
1642 sS  DATA 00,00,0B,08,00,00,0B,22,00,00
1643 1o  DATA 0B,3C,00,00,0B,AA,00,00,0B,B0
1644 1x  DATA 00,00,0B,BE,00,00,0B,C4,00,00
1645 PR  DATA 0B,CE,00,00,0B,E0,00,00,0B,EA
1646 0R  DATA 00,00,0C,00,00,00,0C,06,00,00
1647 k8  DATA 0C,0A,00,00,0C,10,00,00,0C,1A
1648 Iu  DATA 00,00,0C,24,00,00,0C,2C,00,00
1649 Mj  DATA 0C,3A,00,00,0C,44,00,00,0C,52
1650 e8  DATA 00,00,0C,6C,00,00,0C,86,00,00
1651 5U  DATA 0C,A0,00,00,0C,BA,00,00,0C,D4
1652 2Q  DATA 00,00,0C,EE,00,00,0D,08,00,00
1653 2I  DATA 0D,1A,00,00,0D,20,00,00,0D,32
1654 kJ  DATA 00,00,0D,38,00,00,0D,4A,00,00
1655 U5  DATA 0D,64,00,00,0D,6A,00,00,0D,82
1656 IO  DATA 00,00,0D,AC,00,00,0D,BC,00,00
1657 Yj  DATA 0D,CA,00,00,0D,D0,00,00,0D,D6
1658 Fr  DATA 00,00,0D,E4,00,00,0D,EA,00,00
1659 7E  DATA 0D,F0,00,00,0D,F6,00,00,0E,04
1660 Ps  DATA 00,00,0E,10,00,00,0E,1A,00,00
1661 e2  DATA 0E,48,00,00,0E,62,00,00,0E,6E
1662 mt  DATA 00,00,0E,78,00,00,0E,84,00,00
1663 1b  DATA 0E,94,00,00,0E,A4,00,00,0E,B6
1664 Ds  DATA 00,00,0E,C4,00,00,0E,CA,00,00
1665 Tt  DATA 0E,D0,00,00,0E,D6,00,00,0E,E4
1666 uQ  DATA 00,00,0E,EA,00,00,0E,F0,00,00
1667 Jv  DATA 0E,F6,00,00,0F,04,00,00,0F,0A
1668 pJ  DATA 00,00,0F,14,00,00,0F,1A,00,00
1669 On  DATA 0F,28,00,00,0F,2E,00,00,0F,48
1670 ri  DATA 00,00,0F,6C,00,00,10,0C,00,00
1671 Pv  DATA 10,3E,00,00,10,4E,00,00,10,84
1672 7N  DATA 00,00,10,8A,00,00,10,94,00,00
1673 QE  DATA 10,A6,00,00,10,B0,00,00,10,C6
1674 YF  DATA 00,00,10,CC,00,00,10,D0,00,00
1675 ov  DATA 10,D6,00,00,10,E0,00,00,10,EA
1676 SI  DATA 00,00,10,F2,00,00,11,00,00,00
1677 pE  DATA 11,0A,00,00,11,18,00,00,11,32
1678 2W  DATA 00,00,11,4C,00,00,11,66,00,00
1679 gr  DATA 11,80,00,00,11,9A,00,00,11,B4
1680 RG  DATA 00,00,11,CE,00,00,11,E8,00,00
1681 Eq  DATA 11,FA,00,00,12,00,00,00,12,12
1682 8h  DATA 00,00,12,18,00,00,12,2A,00,00
1683 zq  DATA 12,30,00,00,12,42,00,00,12,5C
1684 hv  DATA 00,00,12,62,00,00,12,7A,00,00
1685 ce  DATA 12,A4,00,00,12,B4,00,00,12,C2
1686 11  DATA 00,00,12,C8,00,00,12,CE,00,00
1687 xD  DATA 12,DC,00,00,12,E2,00,00,12,E8
1688 Ub  DATA 00,00,12,EE,00,00,13,00,00,00
1689 zH  DATA 13,0E,00,00,13,18,00,00,13,26
1690 xM  DATA 00,00,13,2C,00,00,13,32,00,00
1691 Bi  DATA 13,3E,00,00,13,44,00,00,13,4A
1692 ju  DATA 00,00,13,50,00,00,13,5A,00,00
1693 Eq  DATA 13,66,00,00,13,70,00,00,13,9A
1694 Sn  DATA 00,00,13,A0,00,00,13,B0,00,00
1695 Wp  DATA 13,CA,00,00,13,D0,00,00,13,E4
1696 iP  DATA 00,00,13,EA,00,00,13,FA,00,00
1697 3g  DATA 14,00,00,00,14,06,00,00,14,16
1698 g1  DATA 00,00,14,24,00,00,14,2A,00,00
1699 Ho  DATA 14,38,00,00,14,3E,00,00,14,48
1700 jI  DATA 00,00,14,4E,00,00,14,5C,00,00
1701 7p  DATA 14,62,00,00,14,70,00,00,14,76
1702 rz  DATA 00,00,14,86,00,00,14,9C,00,00
1703 NX  DATA 14,A6,00,00,14,B0,00,00,14,BA
1704 Z1  DATA 00,00,14,C4,00,00,15,3A,00,00
1705 KD  DATA 15,44,00,00,15,4E,00,00,15,58
1706 rd  DATA 00,00,15,86,00,00,15,90,00,00
1707 Ow  DATA 15,9A,00,00,15,A4,00,00,16,04
1708 26  DATA 00,00,16,16,00,00,16,20,00,00
1709 pQ  DATA 16,36,00,00,16,3C,00,00,16,40
1710 VQ  DATA 00,00,16,46,00,00,16,50,00,00
1711 YI  DATA 16,5A,00,00,16,62,00,00,16,70
1712 qv  DATA 00,00,16,7A,00,00,16,88,00,00
1713 dJ  DATA 16,A2,00,00,16,BC,00,00,16,D6
1714 4r  DATA 00,00,16,F0,00,00,17,0A,00,00
1715 cY  DATA 17,24,00,00,17,3E,00,00,17,58
1716 PM  DATA 00,00,17,6A,00,00,17,70,00,00
1717 Sh  DATA 17,82,00,00,17,88,00,00,17,9A
1718 cn  DATA 00,00,17,A0,00,00,17,B2,00,00
1719 tV  DATA 17,CC,00,00,17,D2,00,00,17,EA
1720 UQ  DATA 00,00,18,14,00,00,18,20,00,00
1721 qX  DATA 18,26,00,00,18,34,00,00,18,3A
1722 Ay  DATA 00,00,18,40,00,00,18,46,00,00
1723 OB  DATA 18,54,00,00,18,62,00,00,18,68
1724 fd  DATA 00,00,18,76,00,00,18,7C,00,00
1725 w0  DATA 18,86,00,00,18,8C,00,00,18,98
1726 JY  DATA 00,00,18,A2,00,00,18,A8,00,00
1727 uZ  DATA 18,B4,00,00,18,C4,00,00,18,CE
1728 Hu  DATA 00,00,18,DE,00,00,18,EE,00,00
1729 qh  DATA 18,F6,00,00,19,06,00,00,19,18
1730 k0  DATA 00,00,19,1E,00,00,19,26,00,00
1731 CF  DATA 19,36,00,00,19,3C,00,00,19,4C
1732 TQ  DATA 00,00,19,54,00,00,19,5A,00,00
1733 Oi  DATA 19,6A,00,00,19,84,00,00,19,8E
1734 V9  DATA 00,00,19,94,00,00,19,A2,00,00
1735 8J  DATA 19,A8,00,00,19,B8,00,00,19,C2
1736 ak  DATA 00,00,19,D0,00,00,19,DA,00,00
1737 Hc  DATA 19,E8,00,00,19,FE,00,00,1A,02
1738 JC  DATA 00,00,1A,08,00,00,1A,1A,00,00
1739 BW  DATA 1A,20,00,00,1A,32,00,00,1A,38
1740 7g  DATA 00,00,1A,4A,00,00,1A,50,00,00
1741 6j  DATA 1A,62,00,00,1A,68,00,00,1A,7A
1742 JR  DATA 00,00,1A,80,00,00,1A,92,00,00
1743 HL  DATA 1A,AC,00,00,1A,BC,00,00,1A,E2
1744 kR  DATA 00,00,1A,E8,00,00,1A,F6,00,00
1745 Ib  DATA 1A,FC,00,00,1B,00,00,00,1B,06
1746 R5  DATA 00,00,1B,10,00,00,1B,1A,00,00
1747 If  DATA 1B,22,00,00,1B,30,00,00,1B,3A
1748 Dr  DATA 00,00,1B,48,00,00,1B,5A,00,00
1749 8f  DATA 1B,64,00,00,1B,7A,00,00,1B,94
1750 17  DATA 00,00,1B,AE,00,00,1B,C8,00,00
1751 au  DATA 1B,E2,00,00,1B,FC,00,00,1C,0E
1752 Af  DATA 00,00,1C,14,00,00,1C,26,00,00
1753 cA  DATA 1C,2C,00,00,1C,3E,00,00,1C,58
1754 4e  DATA 00,00,1C,68,00,00,1C,8E,00,00
1755 U7  DATA 1C,94,00,00,1C,A2,00,00,1C,B0
1756 YS  DATA 00,00,1C,B6,00,00,1C,BC,00,00
1757 ln  DATA 1C,C6,00,00,1C,CA,00,00,1C,D4
1758 Bh  DATA 00,00,1C,E2,00,00,1C,E8,00,00
1759 Pg  DATA 1C,EE,00,00,1C,F8,00,00,1C,FC
1760 Ko  DATA 00,00,1D,02,00,00,1D,08,00,00
1761 HY  DATA 1D,14,00,00,1D,1A,00,00,1D,20
1762 Ir  DATA 00,00,1D,24,00,00,1D,2A,00,00
1763 zQ  DATA 1D,34,00,00,1D,3A,00,00,1D,44
1764 V5  DATA 00,00,1D,4E,00,00,1D,56,00,00
1765 A0  DATA 1D,60,00,00,1D,6A,00,00,1D,74
1766 yN  DATA 00,00,1D,7A,00,00,1D,88,00,00
1767 S5  DATA 1D,96,00,00,1D,B0,00,00,1D,B6
1768 Vc  DATA 00,00,1D,BC,00,00,1D,CE,00,00
1769 o9  DATA 1D,D4,00,00,1D,E6,00,00,1D,EC
1770 4J  DATA 00,00,1D,FE,00,00,1E,04,00,00
1771 Ti  DATA 1E,16,00,00,1E,30,00,00,1E,3A
1772 5J  DATA 00,00,1E,44,00,00,1E,72,00,00
1773 8E  DATA 1E,8C,00,00,1E,A6,00,00,1E,AC
1774 zV  DATA 00,00,1E,B6,00,00,1E,C0,00,00
1775 Di  DATA 1E,CA,00,00,1E,D4,00,00,1E,DE
1776 Kd  DATA 00,00,1E,E2,00,00,1E,F0,00,00
1777 T4  DATA 1E,F6,00,00,1F,00,00,00,1F,1E
1778 sL  DATA 00,00,1F,24,00,00,1F,2A,00,00
1779 kM  DATA 1F,56,00,00,1F,66,00,00,1F,E6
1780 sN  DATA 00,00,1F,F2,00,00,1F,FC,00,00
1781 AH  DATA 20,06,00,00,20,10,00,00,20,1A
1782 g0  DATA 00,00,20,20,00,00,20,26,00,00
1783 ZJ  DATA 20,30,00,00,20,34,00,00,20,3E
1784 Ht  DATA 00,00,20,48,00,00,20,5E,00,00
1785 k5  DATA 20,68,00,00,20,72,00,00,20,7C
1786 m9  DATA 00,00,20,86,00,00,20,8E,00,00
1787 rh  DATA 20,94,00,00,20,9A,00,00,20,A4
1788 OU  DATA 00,00,20,AA,00,00,20,BC,00,00
1789 fh  DATA 20,C6,00,00,20,D0,00,00,20,DA
1790 XD  DATA 00,00,20,E4,00,00,20,EC,00,00
1791 hg  DATA 20,F2,00,00,20,F8,00,00,21,04
1792 sS  DATA 00,00,21,10,00,00,21,2E,00,00
1793 1U  DATA 21,34,00,00,21,40,00,00,21,4C
1794 p6  DATA 00,00,21,6A,00,00,21,70,00,00
1795 8t  DATA 21,7C,00,00,21,88,00,00,21,A6
1796 2w  DATA 00,00,21,AC,00,00,21,B8,00,00
1797 EM  DATA 21,C4,00,00,21,E2,00,00,21,E8
1798 md  DATA 00,00,21,F4,00,00,22,00,00,00
1799 B1  DATA 22,1A,00,00,22,20,00,00,22,2C
1800 Xo  DATA 00,00,22,38,00,00,22,52,00,00
1801 GT  DATA 22,58,00,00,22,62,00,00,22,68
1802 xB  DATA 00,00,22,72,00,00,22,7C,00,00
1803 K7  DATA 22,84,00,00,22,92,00,00,22,9C
1804 vS  DATA 00,00,22,B2,00,00,22,C8,00,00
1805 yv  DATA 22,D6,00,00,22,FA,00,00,23,14
1806 Gm  DATA 00,00,23,2C,00,00,23,46,00,00
1807 4p  DATA 23,4C,00,00,23,5E,00,00,23,64
1808 DB  DATA 00,00,23,76,00,00,23,90,00,00
1809 r1  DATA 23,96,00,00,23,AE,00,00,23,D8
1810 Di  DATA 00,00,23,E4,00,00,23,EA,00,00
1811 ZB  DATA 23,F8,00,00,24,06,00,00,24,0C
1812 Zz  DATA 00,00,24,12,00,00,24,1C,00,00
1813 SL  DATA 24,20,00,00,24,26,00,00,24,2C
1814 Oj  DATA 00,00,24,3C,00,00,24,42,00,00
1815 WA  DATA 24,4E,00,00,24,5A,00,00,24,66
1816 nk  DATA 00,00,24,72,00,00,24,78,00,00
1817 8e  DATA 24,7E,00,00,24,84,00,00,24,92
1818 Zc  DATA 00,00,24,98,00,00,24,A6,00,00
1819 q8  DATA 24,B4,00,00,24,BA,00,00,24,C0
1820 Ga  DATA 00,00,24,C6,00,00,24,D0,00,00
1821 rC  DATA 24,DA,00,00,24,E2,00,00,24,F0
1822 73  DATA 00,00,24,FA,00,00,25,10,00,00
1823 b0  DATA 25,26,00,00,25,34,00,00,25,58
1824 im  DATA 00,00,25,72,00,00,25,8A,00,00
1825 S3  DATA 25,A4,00,00,25,AA,00,00,25,BC
1826 ev  DATA 00,00,25,C2,00,00,25,D4,00,00
1827 r1  DATA 25,EE,00,00,25,F4,00,00,26,0C
1828 WY  DATA 00,00,26,36,00,00,26,42,00,00
1829 7U  DATA 26,48,00,00,26,56,00,00,26,64
1830 Ws  DATA 00,00,26,6A,00,00,26,6E,00,00
1831 PG  DATA 26,74,00,00,26,7A,00,00,26,80
1832 XI  DATA 00,00,26,92,00,00,26,A0,00,00
1833 OY  DATA 26,A6,00,00,26,C6,00,00,26,D4
1834 nY  DATA 00,00,26,DE,00,00,26,EE,00,00
1835 1g  DATA 26,F4,00,00,27,00,00,00,27,0C
1836 1r  DATA 00,00,27,18,00,00,27,24,00,00
1837 1a  DATA 27,2C,00,00,27,32,00,00,27,36
1838 GS  DATA 00,00,27,3C,00,00,27,42,00,00
1839 WO  DATA 27,4C,00,00,27,50,00,00,27,5E
1840 tF  DATA 00,00,27,6E,00,00,27,7C,00,00
1841 Vy  DATA 27,9C,00,00,27,A8,00,00,27,AE
1842 Kd  DATA 00,00,27,B4,00,00,27,D6,00,00
1843 oC  DATA 27,DC,00,00,27,EC,00,00,28,06
1844 hx  DATA 00,00,28,2A,00,00,28,38,00,00
1845 4K  DATA 28,56,00,00,28,8C,00,00,28,AA
1846 8Z  DATA 00,00,28,CC,00,00,28,D6,00,00
1847 PM  DATA 28,E0,00,00,28,EA,00,00,28,F4
1848 pf  DATA 00,00,29,50,00,00,29,A0,00,00
1849 8y  DATA 29,B2,00,00,29,B8,00,00,29,BE
1850 mh  DATA 00,00,29,E0,00,00,29,F2,00,00
1851 wN  DATA 29,F8,00,00,29,FE,00,00,2A,18
1852 c3  DATA 00,00,2A,30,00,00,2A,44,00,00
1853 13  DATA 2A,4C,00,00,2A,66,00,00,2A,78
1854 iE  DATA 00,00,2A,7C,00,00,2A,82,00,00
1855 Rw  DATA 2A,98,00,00,2A,9E,00,00,2A,A6
1856 KO  DATA 00,00,2A,AC,00,00,2A,B4,00,00
1857 lk  DATA 2A,BA,00,00,2A,C2,00,00,2A,C8
1858 OA  DATA 00,00,2A,F4,00,00,2A,FA,00,00
1859 Dc  DATA 2B,22,00,00,2B,48,00,00,2B,96
1860 b5  DATA 00,00,2B,9C,00,00,2B,A2,00,00
1861 VI  DATA 2B,A8,00,00,2B,AE,00,00,2B,B4
1862 Gu  DATA 00,00,2B,C4,00,00,2B,D2,00,00
1863 9G  DATA 2B,DA,00,00,2B,E6,00,00,2B,F2
1864 Eo  DATA 00,00,2B,FE,00,00,2C,0A,00,00
1865 40  DATA 2C,12,00,00,2C,18,00,00,2C,38
1866 mI  DATA 00,00,2C,64,00,00,2C,6E,00,00
1867 K3  DATA 2C,82,00,00,2C,8C,00,00,2C,92
1868 ar  DATA 00,00,2C,98,00,00,2C,B2,00,00
1869 gx  DATA 2C,BA,00,00,2C,CC,00,00,2C,D2
1870 pn  DATA 00,00,2C,F2,00,00,2D,10,00,00
1871 pC  DATA 2D,14,00,00,2D,1C,00,00,2D,26
1872 ur  DATA 00,00,2D,2A,00,00,2D,D8,00,00
1873 Op  DATA 2D,F6,00,00,2E,08,00,00,2E,18
1874 Co  DATA 00,00,2E,2A,00,00,2E,46,00,00
1875 rZ  DATA 2E,4A,00,00,2E,88,00,00,2E,98
1876 PI  DATA 00,00,2E,AA,00,00,2E,C6,00,00
1877 mh  DATA 2E,CA,00,00,2F,3E,00,00,2F,50
1878 4J  DATA 00,00,2F,60,00,00,2F,6A,00,00
```

Listing 1. (Fortsetzung)

```
VV DATA 2F,6E,00,00,2F,9A,00,00,2F,AC
nb DATA 00,00,2F,BC,00,00,2F,C6,00,00
ED DATA 2F,CA,00,00,30,08,00,00,30,18
Us DATA 00,00,30,22,00,00,30,26,00,00
KQ DATA 30,58,00,00,30,68,00,00,30,72
bu DATA 00,00,30,76,00,00,30,A2,00,00
Y1 DATA 30,B4,00,00,30,C4,00,00,30,CE
x0 DATA 00,00,30,D2,00,00,31,10,00,00
3y DATA 31,20,00,00,31,2A,00,00,31,2E
YD DATA 00,00,31,5A,00,00,31,6C,00,00
3m DATA 31,7C,00,00,31,86,00,00,31,8A
P5 DATA 00,00,31,B6,00,00,31,C8,00,00
F4 DATA 31,D8,00,00,31,E2,00,00,31,E6
xG DATA 00,00,32,24,00,00,32,34,00,00
xg DATA 32,3E,00,00,32,42,00,00,33,DE
vF DATA 00,00,33,E2,00,00,33,E6,00,00
Fr DATA 33,EA,00,00,33,EE,00,00,33,F2
wS DATA 00,00,33,F6,00,00,33,FA,00,00
Ad DATA 33,FE,00,00,34,02,00,00,34,06
YK DATA 00,00,34,0A,00,00,34,0E,00,00
tX DATA 34,12,00,00,34,16,00,00,34,1A
gB DATA 00,00,34,1E,00,00,34,22,00,00
L1 DATA 34,26,00,00,34,2A,00,00,34,2E
eo DATA 00,00,34,32,00,00,34,46,00,00
rn DATA 34,5A,00,00,34,6E,00,00,34,82
A8 DATA 00,00,34,AA,00,00,34,BE,00,00
Sn DATA 34,D2,00,00,34,E6,00,00,35,0E
jt DATA 00,00,35,22,00,00,35,36,00,00
uq DATA 35,4A,00,00,35,5E,00,00,35,72
7Q DATA 00,00,35,86,00,00,35,AE,00,00
zd DATA 35,D6,00,00,35,EA,00,00,35,FE
oy DATA 00,00,36,12,00,00,36,26,00,00
Eu DATA 00,1D,00,00,00,01,00,00,0E,2C
hE DATA 00,00,0E,3E,00,00,0E,54,00,00
Gm DATA 0E,68,00,00,0E,AC,00,00,0F,40
ag DATA 00,00,0F,54,00,00,0F,64,00,00
Ea DATA 0F,78,00,00,0F,82,00,00,0F,A0
Qu DATA 00,00,0F,B8,00,00,0F,D0,00,00
D7 DATA 13,7E,00,00,13,86,00,00,13,AA
wH DATA 00,00,13,B6,00,00,15,B6,00,00
bQ DATA 15,EE,00,00,1E,B0,00,00,1E,BA
jU DATA 00,00,1E,C4,00,00,1E,CE,00,00
p4 DATA 1E,D8,00,00,1F,EE,00,00,1F,F8
YD DATA 00,00,20,02,00,00,20,0C,00,00
Qm DATA 20,16,00,00,00,00,00,00,03,F2
w7 DATA 00,00,03,EB,00,00,01,00,00,00
qX DATA 03,F2
(C) 1987 M&T
```

Listing 1. (Schluß)

Der Konvertierer

Von Andreas Lietz

Mit dem »Amiga Object Editor« aus dem Sonderheft 2 konnte man zwar wunderschöne Sprites entwerfen, aber sie ließen sich nicht mit Amiga-Basic verwenden. Mit »AOE-Convert« bauen Sie diese Objekte problemlos in Ihre Basic-Programme ein.

Haben Sie schon einmal mit dem langsamen und umständlichen Amiga-Basic-Objekteditor gearbeitet? Dann werden Sie es sehr zu schätzen wissen, daß Sie jetzt die Objekte Ihrer Träume komfortabel mit dem »Amiga Object Editor« (AOE) aus dem vorangehenden Artikel (Seite 6).

»AOE-Convert« wandelt sie in das BOB-Format, auf das Sie mit Hilfe der OBJECT-Befehle des Amiga-Basic bequem Zugriff haben.

Da AOE-Convert in Assembler geschrieben ist, haben wir das Programm als Basic-Lader (Listing 1) abgedruckt. Assembler-Interessierte finden in Listing 2 den Quellcode des Programms.

Tippen Sie bitte Listing 1 mit Hilfe des Checksummers auf Seite 159 ab und speichern es als »AOE-Convert.bas«. Nach dem Start mit »RUN« erzeugt der Basic-Lader das lauffähige Programm »AOE-Convert« auf Diskette.

Dieses Programm rufen Sie durch Eingabe seines Namens aus dem CLI auf; Sie können ihm auch mit dem CLI-Befehl

```
copy icon.info
AOE-Convert.info
```

ein Icon zuweisen und das Programm über die Workbench aufrufen.

Die Anwendung des Programms ist ganz einfach: Nachdem Sie es gestartet haben, erscheint am oberen Bildschirmrand ein Fenster (Bild 1), in dem nach der AOE-Datei gefragt wird. Geben Sie also den Namen der Datei ein, die Sie konvertieren wollen.

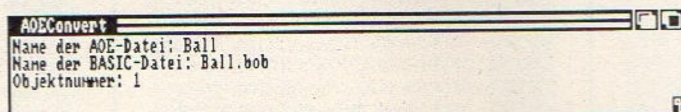


Bild 1. Das Fenster von »AOE-Convert«

Nun fragt das Programm nach der Datei, in die das konvertierte Objekt geschrieben werden soll.

Als letztes erfolgt die Frage nach der Objektnummer. Amiga-Basic kann nämlich nur einzelne BOBs in einer Datei verarbeiten - AOE speichert aber unter Umständen mehrere Objekte zusammen auf Diskette.

Wenn Sie auf diese Frage mit <RETURN> antworten, wird das erste Objekt der Datei konvertiert. Falls das Programm nach dieser Frage kommentarlos abbricht, ist ein Fehler aufgetreten.

Möglicherweise wurde die Quelldatei nicht gefunden oder sie war keine AOE-Datei. Auch beim Schreiben der Zieldatei kann etwas nicht funktioniert haben.

Nachdem das Objekt konvertiert wurde, fragt das Programm nach dem Namen eines weiteren Objekts. Ist alles konvertiert, verlassen Sie »AOE-Convert«, indem Sie <RETURN> ohne zusätzliche Eingabe drükken.

Achtung: Sie sollten nach dem Konvertieren die AOE-Datei nicht löschen! Vielleicht möchten Sie ja später noch einmal an dem Objekt arbeiten. Außerdem wird diese Datei für die richtige Einstellung der Farben im Basic-Programm benötigt.

Die konvertierten BOBs lassen sich ganz normal mit dem OBJECT.SHAPE-Befehl von Amiga-Basic laden. Sie werden in der richtigen Größe angezeigt, wenn Sie einen niedrigauflösenden Screen (320 x 256 Punkte) mit 16 Farben verwenden. Damit das Objekt die richtigen Farben bekommt, müßte man mit Hilfe des Palette-Befehls nun mühsam die Farben einstellen, mit denen das Objekt zuvor erzeugt wurde.

Diese Arbeit nimmt Ihnen das kleine Programm in Listing 3 ab. Es macht sich die Tatsache zunutze, daß der Amiga Object Editor in seinen Dateien die Farben des Objektes mit speichert.

Bauen Sie Listing 3 an einer Stelle Ihres Programms ein, an der der Screen und das Fenster bereits geöffnet wurden, in denen sich Ihr Objekt bewegen soll. Wenn Sie in den OPEN-Befehl in Zeile 2 den Namen Ihrer AOE-Datei einsetzen, werden die Farben automatisch richtig eingestellt. Ein Beispiel, bei dem das Objekt auf dem Bildschirm bewegt wird, sehen Sie in Listing 4.

Natürlich müssen Sie auch hier in Zeile 4 und 15 die OPEN-Befehle entsprechend anpassen.

Damit haben Sie von Basic aus Zugriff auf die Dateien des »Amiga Object Editor« (AOE) aus dieser Ausgabe (Seite 6).

Sie stellen mit AOE einen BOB her, bringen das Ergebnis mit Hilfe von AOE-Convert auf Basic-Format und binden diese Datei als BOB in Ihre eigenen Programme ein. *kn*

Programmname:	AOE-Convert.bas
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Amiga-Basic
Bemerkung:	Basiclader für »AOE-Convert«

```
Programmautor: Michael Bertsch
------------------------------

OmO REM Generiert lauffähiges Programm
ag  CLS
KJ  OPEN "AOE-Convert" FOR OUTPUT AS 1
```

```
4 BS   READ anz
5 oa   FOR i=1 TO anz
6 3n1   READ h$
7 yB2    wert1=ASC(LEFT$(h$,1))
8 bP     IF wert1>64 THEN wert1=wert1-87
         ELSE wert1=wert1-48
9 FI     wert1=wert1*16
10 7c    wert2=ASC(RIGHT$(h$,1))
11 wp    IF wert2>64 THEN wert2=wert2-87
         ELSE wert2=wert2-48
12 Pi    wert=wert1+wert2
13 9G    PRINT #1,CHR$(wert);
14 JOO NEXT
15 3n  CLOSE 1
16 Ov  END
17 yc  Werte:
18 SA  DATA 1092
19 ph  DATA 00,00,03,f3,00,00,00,00,00,00
20 ms  DATA 00,03,00,00,00,00,00,00,00,02
21 IL  DATA 00,00,00,aa,00,00,00,19,00,00
22 k8  DATA 00,ac,00,00,03,e9,00,00,00,aa
23 lz  DATA 2c,78,00,04,93,c9,4e,ae,fe,da
24 PH  DATA 28,40,4a,ac,00,ac,66,16,41,ec
25 Xd  DATA 00,5c,4e,ae,fe,80,41,ec,00,5c
26 f9  DATA 4e,ae,fe,8c,23,c0,00,00,00,00
27 23  DATA 43,f9,00,00,00,00,70,00,4e,ae
28 yq  DATA fd,d8,23,c0,00,00,00,04,67,00
29 Mc  DATA 01,e4,2c,79,00,00,00,04,22,3c
30 re  DATA 00,00,00,0c,24,3c,00,00,03,ed
31 EA  DATA 4e,ae,ff,e2,23,c0,00,00,00,08
32 eI  DATA 67,00,01,c4,22,39,00,00,00,08
33 JV  DATA 24,3c,00,00,00,28,61,00,02,2c
34 vF  DATA 4e,ae,ff,d0,22,39,00,00,00,08
35 QP  DATA 24,3c,00,00,00,0c,76,32,4e,ae
36 mY  DATA ff,d6,53,00,67,00,01,98,41,f9
37 wG  DATA 00,00,00,0c,42,30,00,00,22,39
38 HW  DATA 00,00,00,08,24,3c,00,00,00,3d
39 JC  DATA 61,00,01,f6,4e,ae,ff,d0,22,39
40 pN  DATA 00,00,00,08,24,3c,00,00,00,48
41 zH  DATA 76,32,4e,ae,ff,d6,53,00,67,00
42 XX  DATA 01,62,41,f9,00,00,00,48,42,30
43 er  DATA 00,00,22,39,00,00,00,08,24,3c
44 t0  DATA 00,00,00,54,61,00,01,c0,4e,ae
45 E1  DATA ff,d0,22,39,00,00,00,08,24,3c
46 tc  DATA 00,00,00,86,76,04,4e,ae,ff,d6
47 2c  DATA 53,00,66,04,7e,01,60,32,41,f9
48 DP  DATA 00,00,00,86,41,f0,00,00,7e,00
49 eT  DATA 7c,00,1e,20,67,20,9e,3c,00,30
50 9f  DATA 1c,20,67,18,9c,3c,00,30,cc,fc
51 TW  DATA 00,0a,de,46,1c,20,67,0a,9c,3c
52 F4  DATA 00,30,cc,fc,00,64,de,46,33,c7
53 n0  DATA 00,00,00,86,22,3c,00,00,00,0c
54 Zz  DATA 24,3c,00,00,03,ed,4e,ae,ff,e2
55 6p  DATA 23,c0,00,00,00,8c,67,00,00,d8
56 tH  DATA 22,39,00,00,00,8c,24,3c,00,00
57 wv  DATA 00,94,76,26,4e,ae,ff,d6,41,f9
58 7c  DATA 00,00,00,94,0c,90,41,4f,45,20
59 vS  DATA 66,00,00,b6,22,39,00,00,00,8c
60 b7  DATA 34,39,00,00,00,86,53,42,c4,fc
61 j4  DATA 02,80,76,00,4e,ae,ff,be,22,39
62 L5  DATA 00,00,00,8c,24,3c,00,00,00,94
63 lY  DATA 26,3c,00,00,02,00,4e,ae,ff,d6
64 in  DATA 22,39,00,00,00,8c,4e,ae,ff,dc
65 h0  DATA 42,b9,00,00,00,8c,22,3c,00,00
66 GY  DATA 00,48,24,3c,00,00,03,ee,4e,ae
67 N1  DATA ff,e2,23,c0,00,00,00,90,67,00
68 yl  DATA 00,5e,41,f9,00,00,02,94,42,98
69 SF  DATA 42,98,20,fc,00,00,00,04,7a,20
70 qq  DATA 20,c5,20,c5,30,fc,00,18,30,fc
71 qv  DATA 00,0f,42,58,22,39,00,00,00,90
72 8G  DATA 24,3c,00,00,02,94,76,1a,4e,ae
73 bP  DATA ff,d0,22,39,00,00,00,90,24,3c
74 RD  DATA 00,00,00,94,26,3c,00,00,02,00
75 Wr  DATA 4e,ae,ff,d0,22,39,00,00,00,90
76 pm  DATA 4e,ae,ff,dc,42,b9,00,00,00,90
77 hT  DATA 60,00,fe,40,4a,b9,00,00,00,90
78 fr  DATA 67,10,2c,79,00,00,00,04,22,39
79 8u  DATA 00,00,00,90,4e,ae,ff,dc,4a,b9
80 1V  DATA 00,00,00,8c,67,10,2c,79,00,00
81 1v  DATA 00,04,22,39,00,00,00,8c,4e,ae
82 gs  DATA ff,dc,4a,b9,00,00,00,08,67,10
83 a0  DATA 2c,79,00,00,00,04,22,39,00,00
84 Fr  DATA 00,08,4e,ae,ff,dc,4a,b9,00,00
85 19  DATA 00,04,67,0e,2c,78,00,04,22,79
86 1y  DATA 00,00,00,04,4e,ae,fe,62,4a,b9
87 CZ  DATA 00,00,00,00,67,0e,2c,78,00,04
88 Cw  DATA 22,79,00,00,00,00,4e,ae,fe,86
89 Ik  DATA 70,00,4e,75,20,42,76,00,4a,18
90 F2  DATA 67,04,52,83,60,f8,4e,75,00,00
91 1Y  DATA 00,00,03,ec,00,00,00,05,00,00
92 1w  DATA 00,01,00,00,00,2a,00,00,00,46
93 Ow  DATA 00,00,00,66,00,00,00,9c,00,00
94 GL  DATA 00,d2,00,00,00,32,00,00,00,02
95 Oj  DATA 00,00,00,24,00,00,00,36,00,00
96 iJ  DATA 00,40,00,00,00,56,00,00,00,60
97 se  DATA 00,00,00,74,00,00,00,7a,00,00
98 rm  DATA 00,8c,00,00,00,96,00,00,00,aa
99 2v  DATA 00,00,00,b0,00,00,00,c2,00,00
100 Hs DATA 00,cc,00,00,00,e0,00,00,00,e6
101 vH DATA 00,00,00,fa,00,00,01,2c,00,00
102 q3 DATA 01,32,00,00,01,42,00,00,01,4c
103 2t DATA 00,00,01,52,00,00,01,5e,00,00
104 sG DATA 01,6e,00,00,01,74,00,00,01,86
105 OU DATA 00,00,01,8c,00,00,01,9c,00,00
106 nk DATA 01,a6,00,00,01,ac,00,00,01,bc
107 yw DATA 00,00,01,c6,00,00,01,e6,00,00
108 Ob DATA 01,ec,00,00,01,f8,00,00,01,fe
109 MW DATA 00,00,02,0e,00,00,02,18,00,00
110 5D DATA 02,22,00,00,02,2a,00,00,02,30
111 JA DATA 00,00,02,3a,00,00,02,42,00,00
112 le DATA 02,48,00,00,02,52,00,00,02,5a
113 Bq DATA 00,00,02,60,00,00,02,6a,00,00
114 CM DATA 02,76,00,00,02,80,00,00,02,8c
115 tr DATA 00,00,00,00,00,00,03,f2,00,00
116 Km DATA 03,ea,00,00,00,19,64,6f,73,2e
117 OJ DATA 6c,69,62,72,61,72,79,00,43,4f
118 Xp DATA 4e,3a,30,2f,30,2f,36,34,30,2f
119 X3 DATA 35,30,2f,20,41,4f,45,43,6f,6e
120 g6 DATA 76,65,72,74,20,00,4e,61,6d,65
121 4d DATA 20,64,65,72,20,41,4f,45,2d,44
122 fB DATA 61,74,65,69,3a,20,00,4e,61,6d
123 tL DATA 65,20,64,65,72,20,42,41,53,49
124 bz DATA 43,2d,44,61,74,65,69,3a,20,00
125 af DATA 4f,62,6a,65,6b,74,6e,75,6d,6d
126 TY DATA 65,72,3a,20,00,00,00,00,03,f2
127 ex DATA 00,00,03,eb,00,00,00,ac,00,00
128 Re DATA 03,f2
(C) 1989 M&T
```

Listing 1. Der Basic-Lader für »AOE-Convert«

Programmname:	AOE-Convert
Computer:	A500, A1000, A2000 mit Kickstart 1.2 oder 1.3
Sprache:	Assembler
Bemerkung:	-

```
;   AOEConvert
;     9.2.1989
; Michael Bertsch

ExecBase        = 4
FindTask        = -294
WaitPort        = -384
GetMsg          = -372
ReplyMsg        = -378
CloseLibrary    = -414
OpenLibrary     = -552
Open            = -30
Close           = -36
Read            = -42
Write           = -48
Seek            = -66
current         = 0
readOnly        = 1005
newFile         = 1006

Move.l ExecBase,A6
Sub.l A1,A1
Jsr FindTask(A6)
Move.l D0,A4
Tst.l $ac(A4)
Bne.s Start
Lea $5c(A4),A0
Jsr WaitPort(A6)
Lea $5c(A4),A0
```

Listing 2. Für Assembler-Interessierte: das Quellprogramm von »AOE-Convert«

```
 Jsr GetMsg(A6)
 Move.l D0,WBMsg
Start:
 Lea DOSName,A1                  ; DOS-Library öffnen
 Moveq #0,D0
 Jsr OpenLibrary(A6)
 Move.l D0,DOSBase
 Beq Ende
 Move.l DOSBase,A6
 Move.l #Fenster,D1              ; Fenster öffnen
 Move.l #readOnly,D2
 Jsr Open(A6)
 Move.l D0,FensterHandle
 Beq Ende

Schleife:
 Move.l FensterHandle,D1         ; Frage nach dem 1. Dateinamen
 Move.l #AOEFileFrage,D2
 Bsr Länge
 Jsr Write(A6)
 Move.l FensterHandle,D1         ; 1. Dateinamen holen
 Move.l #AOEFileName,D2
 Moveq #50,D3
 Jsr Read(A6)
 Subq.b #1,D0
 Beq Ende
 Lea AOEFileName,A0
 Clr.b (A0,D0)
 Move.l FensterHandle,D1         ; Frage nach dem 2. Dateinamen
 Move.l #BASICFileFrage,D2
 Bsr Länge
 Jsr Write(A6)
 Move.l FensterHandle,D1         ; 2. Dateinamen holen
 Move.l #BASICFileName,D2
 Moveq #50,D3
 Jsr Read(A6)
```

```
 Subq.b #1,D0
 Beq Ende
 Lea BASICFileName,A0
 Clr.b (A0,D0)
 Move.l FensterHandle,D1      ; Frage nach der Objektnummer
 Move.l #ObjektnummerFrage,D2
 Bsr Länge
 Jsr Write(A6)
 Move.l FensterHandle,D1      ; Objektnummer holen
 Move.l #Nummer,D2
 Moveq #4,D3
 Jsr Read(A6)
 Subq.b #1,D0
 Bne.s Rechnen
 Moveq #1,D7
 Bra.s Weiter

Rechnen:
 Lea Nummer,A0                ; Objektnummer errechnen
 Lea (A0,D0),A0
 Moveq #0,D7
 Moveq #0,D6
 Move.b -(A0),D7
 Beq.s Weiter
 Sub.b #48,D7
 Move.b -(A0),D6
 Beq.s Weiter
 Sub.b #48,D6
 Mulu #10,D6
 Add.w D6,D7
 Move.b -(A0),D6
 Beq.s Weiter
 Sub.b #48,D6
 Mulu #100,D6
 Add.w D6,D7

Weiter:
 Move.w D7,Nummer
 Move.l #AOEFileName,D1       ; AOE-Datei öffnen
 Move.l #readOnly,D2
 Jsr Open(A6)
 Move.l D0,AOEFileHandle
 Beq Ende
 Move.l AOEFileHandle,D1      ; Objekt einlesen
 Move.l #Puffer,D2
 Moveq #38,D3
 Jsr Read(A6)
 Lea Puffer,A0
 Cmp.l #"AOE ",(A0)
 Bne Ende
 Move.l AOEFileHandle,D1
 Move.w Nummer,D2
 Subq.w #1,D2
 Mulu #640,D2
 Moveq #current,D3
 Jsr Seek(A6)
 Move.l AOEFileHandle,D1
 Move.l #Puffer,D2
 Move.l #512,D3
 Jsr Read(A6)
 Move.l AOEFileHandle,D1      ; AOE-Datei schließen
 Jsr Close(A6)
 Clr.l AOEFileHandle
 Move.l #BASICFileName,D1     ; BASIC-Datei öffnen
 Move.l #newFile,D2
 Jsr Open(A6)
 Move.l D0,BASICFileHandle
 Beq Ende
 Lea Puffer2,A0               ; BASIC-Datei erstellen
 Clr.l (A0)+
 Clr.l (A0)+
 Move.l #4,(A0)+
 Moveq #32,D5
 Move.l D5,(A0)+
 Move.l D5,(A0)+
 Move.w #24,(A0)+
 Move.w #15,(A0)+
 Clr.w (A0)+
 Move.l BASICFileHandle,D1
 Move.l #Puffer2,D2
 Moveq #26,D3
 Jsr Write(A6)
 Move.l BASICFileHandle,D1
 Move.l #Puffer,D2
 Move.l #512,D3
 Jsr Write(A6)
 Move.l BASICFileHandle,D1    ; BASIC-Datei schließen
 Jsr Close(A6)
 Clr.l BASICFileHandle
 Bra Schleife

Ende:
 Tst.l BASICFileHandle
 Beq.s BASICFileZu
 Move.l DOSBase,A6
 Move.l BASICFileHandle,D1    ; BASIC-Datei schließen
 Jsr Close(A6)

BASICFileZu:
 Tst.l AOEFileHandle
 Beq.s AOEFileZu
 Move.l DOSBase,A6
 Move.l AOEFileHandle,D1      ; AOE-Datei schließen
 Jsr Close(A6)

AOEFileZu:
 Tst.l FensterHandle
 Beq.s KeinFenster
 Move.l DOSBase,A6
 Move.l FensterHandle,D1      ; Fenster schließen
 Jsr Close(A6)

KeinFenster:
 Tst.l DOSBase
 Beq.s KeineDOSLibrary
 Move.l ExecBase,A6
 Move.l DosBase,A1            ; DOS-Library schließen
 Jsr CloseLibrary(A6)

KeineDOSLibrary:
 Tst.l WBMsg
 Beq.s KeineWBMsg
 Move.l ExecBase,A6
 Move.l WBMsg,A1
 Jsr ReplyMsg(A6)

KeineWBMsg:
 Moveq #0,D0
 Rts

Länge:
 Move.l D2,A0                 ; Länge des Textes ermitteln
 Moveq #0,D3

L01:
 Tst.b (A0)+
 Beq.s L02
 Addq.l #1,D3
 Bra.s L01

L02:
 Rts
 Data

DOSName: Dc.b "dos.library",0
Fenster: Dc.b "CON:0/0/640/50/ AOEConvert ",0
AOEFileFrage: Dc.b "Name der AOE-Datei: ",0
BASICFileFrage: Dc.b "Name der BASIC-Datei: ",0
ObjektnummerFrage: Dc.b "Objektnummer: ",0

 Bss
 Align

WBMsg: Dc.l 0
DOSBase: Dc.l 0
FensterHandle: Dc.l 0
AOEFileName: Blk.w 30
BASICFileName: Blk.w 30

 Dc.w 0

Nummer: Blk.w 3
AOEFileHandle: Dc.l 0
BASICFileHandle: Dc.l 0
Puffer: Blk.l 128
Puffer2: Blk.w 13

 End      (c) 1989 M&T
```

Listing 2. (Schluß)

Programmname:	Farb-Lader
Computer:	A500, A1000, A2000 mit Kickstart 1.2, 1.3
Sprache:	Amiga-Basic
Bemerkung:	Zum Einbau in eigene Programme

```
Programmautor: Andreas Lietz
----------------------------

' Einlesen der AOE-Datei und Einstellung der Bildschirmfarben
OPEN "AOE-Datei" FOR INPUT AS 1   'Hier Namen der AOE-Datei
einsetzen
a$ = INPUT$(6,#1) ' Dateiheader ueberlesen
FOR i= 0 TO 15     ' Farbdaten lesen
  a$ = INPUT$(1,#1)
  b$ = INPUT$(1,#1)
  PALETTE i,(ASC(a$) AND 15)/16,(ASC(b$) AND 240)/256,(ASC (b
  $) AND 15)/16
NEXT i
CLOSE 1
(C) 1989 M&T
```

Listing 3. Dieses Programm stellt die richtigen Farben für ein Objekt ein

Programmname:	BOB-Move
Computer:	A500, A1000, A2000 mit Kickstart 1.2, 1.3
Sprache:	Amiga-Basic
Bemerkung:	Bitte richtige Dateinamen einsetzen!

```
Programmautor: Andreas Lietz
----------------------------

SCREEN 1,320,256,4,1
WINDOW 4,"AOE-Convert",(30,30)-(150,100),15,1
'Einlesen der AOE-Datei und Einstellung der Bildschirmfarben
OPEN "AOE-Datei" FOR INPUT AS 1   'Hier Namen der AOE-Datei
einsetzen
a$ = INPUT$(6,#1) ' Dateiheader ueberlesen
FOR i= 0 TO 15    ' Farbdaten lesen
  a$ = INPUT$(1,#1)
  b$ = INPUT$(1,#1)
  PALETTE i,(ASC(a$) AND 15)/16,(ASC(b$) AND 240)/256,(ASC (
  b$) AND 15)/16
NEXT i
CLOSE 1
'Bewegen eines Objektes auf dem Bildschirm
ON COLLISION GOSUB Aufprallen
COLLISION ON
OPEN "BOB-Datei" FOR INPUT AS 1   'Hier Namen der konvertier
ten Datei einsetzen
OBJECT.SHAPE 1,INPUT$(LOF(1),1)
CLOSE 1
OBJECT.X 1,10
OBJECT.Y 1,20
OBJECT.VX 1,30
OBJECT.VY 1,30
OBJECT.ON
OBJECT.START
WHILE 1
SLEEP
WEND
Aufprallen:
  x = WINDOW(1)
  WINDOW 4
  i = COLLISION(0)
  IF i=0 THEN RETURN
  j = COLLISION (i)
  IF j=-2 OR j = -4 THEN
    OBJECT.VX i,-OBJECT.VX(i)
  ELSE
    OBJECT.VY i,-OBJECT.VY(i)
  END IF
  OBJECT.START
  WINDOW x
RETURN
(C) 1989 M&T
```

Listing 4. Ein Beispielprogramm, das mit konvertierten Objekten arbeitet

Zauberei mit Bildern

Ob in Stein gemeißelt oder nur aus Umrissen bestehend. »IFF_Filter« macht aus Ihren Farbbildern neue fantastische Grafiken. Probieren Sie es doch einfach einmal aus.

Von Peter Sparlinek

Den Begriff Textverarbeitung kennt jeder. Doch genau wie man Text verarbeiten kann, ist dies auch bei Bildern möglich. Bildverarbeitung ist ein interessantes Thema. Tolle Effekte lassen sich damit erreichen. Doch wie geht das? »IFF_Filter« zeigt es Ihnen. Sehen Sie sich doch einmal die Beispielbilder an. Bild 1 ist das Original, das wir mit verschiedenen Filtern behandelt haben. Die Ergebnisse sehen Sie in Bild 2 bis 4.

Filter spielen in der Bildverarbeitung eine große Rolle. Aus »normalen«, farbigen Bildern können Bilder mit Grautönen gemacht werden, die ganz anders aussehen. Es gibt die verschiedensten Typen wie den Tiefpaßfilter, den Gradientenfilter, etc. Jeder dieser Filter hat eine andere Aufgabe. Der Tiefpaßfilter filtert, wie der Name schon sagt, die hohen Frequenzen aus einem Bild. Unter hohen Frequenzen versteht man bei Bildern den häufigen Wechsel von Farben. Nebenbei entsteht der Effekt der Unschärfe. Je größer der Filter, desto stärker wird die Unschärfe. Mit dem Gradientenfilter kann man sehr gut die Kanten von den Objekten des Bildes hervorheben, während Flächen mit gleicher Intensität (gleicher Farbe) dabei verschwinden. Man erhält also im neuen Bild die Farbübergänge.

Peter Sparlinek



Den ersten Kontakt zu einem Commodore PET 2001 hatte Peter Sparlinek im Jahr 1980. Nachdem Basic zu langsam wurde, erfolgte der Umstieg auf 6502-Assembler. Die nächsten Schritte waren ein VC20 (1983) und ein C64 (1985). Zwei Jahre später wurde ein Amiga 500 angeschafft, auf dem der Autor vornehmlich in C programmiert. Den Anstoß zu IFF_Filter erhielt Peter Sparlinek in seinem Informatikstudium.



Bild 1. Das Originalbild mit einer Auflösung von 320 x 200 Punkten und 32 Farben

Filter – was ist das nun genau und wie funktioniert es? Diese Frage soll am Beispiel des 5x3-Tiefpaßfilters beantwortet werden. Der Tiefpaß berechnet den durchschnittlichen Grauwert in einer definierbaren (hier 5x3) Umgebung. Man kann ein Bild in Form einer Matrix darstellen, deren Elemente die Grauwerte beziehungsweise die Farbe enthalten. Ebenso kann man den 5x3-Tiefpaßfilter als Matrix darstellen.

Die Breite der Filtermatrix ist hier 5, die Höhe 3. Weil es ein Tiefpaßfilter sein soll, haben alle Elemente den Wert 1. Der zentrale Punkt der Filtermatrix ist der Mittelpunkt. Jetzt wird die Filtermatrix auf das Bild gelegt und alle Grauwerte des Bildes werden mit dem entsprechenden Wert der Filtermatrix multipliziert und die Produkte anschließend aufsummiert. Die Summe wird anschließend durch die Summe der Filtermatrixelemente (hier 15) dividiert und schon hat man den Grauwert, den das neue Bild an der Stelle unter dem zentralen Punkt annimmt. Dieses Verfahren wird für alle Bildpunkte angewandt.

Der Inhalt der Filtermatrixelemente legt fest, um was für einen Filter es sich handelt. Eine Übersicht über einige Filter sehen Sie in der Tabelle.

Wie man dem Namen nach schon vermuten kann, bearbeitet IFF_Filter Bilder, die im IFF-Standard vorliegen. Praktisch alle Malprogramme auf dem Amiga verwenden dieses Format. Dabei spielt es keine Rolle, ob ein Pinsel (Brush) oder ein Overscan-Bild (besonderes Grafikformat des Amiga) verarbeitet wird. Eine Ausnahme sind Bilder im HAM-Format, sie können nicht gefiltert werden.

IFF_Filter liefert als Resultat wieder ein IFF-Bild, man kann also die erzeugten Bilder mit den üblichen Programmen weiterbearbeiten. IFF_Filter liest ein Bild ein, wandelt es in ein Graubild (16 Farben) um, läßt den Filter über das Bild laufen und speichert es auf Diskette. Bei 512 KByte Speicher kann es zu Speicherproblemen kommen, wenn ein Bild mit 640x512 Punkten und 16 Farben verarbeitet werden soll.

Der Aufruf im CLI sieht folgendermaßen aus:

```
IFF_Filter <Dateiname>
[Parameter...]
```

Hierbei ist <Dateiname> der Name des IFF-Bildes (gegebenenfalls mit Pfad). Es existieren folgende Parameter:

-d<Dateiname>

Das erzeugte Bild wird nach <Dateiname> geschrieben. Wenn diese Option fehlt, hängt IFF_Filter ».Filter« als Namenserweiterung (Suffix) an, gegebenenfalls auch die Filtergröße.

+i

Während des Programmlaufs zeigt IFF_Filter die aktuelle Funktion an.

-n

Das erzeugte Bild wird negativ (nicht bei LOW und OWN).

-p

Das erzeugte Bild wird ungepackt gespeichert. Die Voreinstellung ist gepackte Speicherung. Sollte ein von IFF_Filter generiertes Bild nicht von Ihrem Malprogramm geladen werden, probieren Sie diesen Parameter.

-h<X>

Für <X> sind Werte von 1 bis 4 erlaubt. Das Eingabebild wird aufgehellt. Bei dem Wert 1 verwendet IFF_Filter normale Helligkeit. Setzen Sie 4 ein, wird das Bild stark aufgehellt.

-<typ>

Hiermit geschieht die Auswahl des Filters. Für <typ> können Sie folgende Parameter eingeben:

ID – Nur Umwandlung in Grautöne
REL1 – Relieffilter vertikal
REL2 – Relieffilter diagonal
GRA1 – Gradientenfilter diagonal
GRA2 – Gradientenfilter vertikal
GRA3 – Gradientenfilter alle 8 Richtungen
LOW – Tiefpaßfilter
OWN – selbst zu definierender Filter

Ein Beispiel verdeutlicht die möglichen Aufrufe:

```
IFF_Filter Seascape -h3 -p -REL1
```

Das IFF-Bild »Seascape« wird geladen und in ein Graubild umgewandelt. Dabei wird um den Faktor 3 aufgehellt. Nach dem Tief-

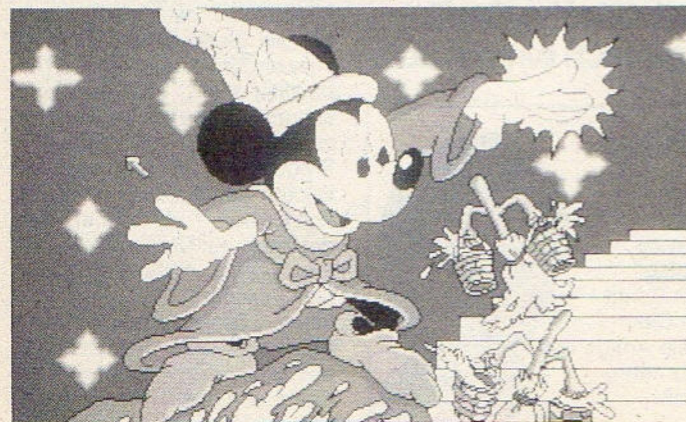


Bild 2. Das Originalbild (siehe Bild 1) nach der Umwandlung in Grautöne ohne Filter



Bild 3. Das Originalbild (siehe Bild 1) bearbeitet mit dem Relieffilter 2 und Aufhellung um den Faktor 4

paß wird das gefilterte Bild ungepackt unter dem Namen »Seascape.Filter« gespeichert.

Natürlich sind auch Kombinationen der verschiedenen Filterarten verwendbar:

```
IFF_Filter DH0:Pic +i -dRAM:Bild -REL1 -GRA2
```

Diesmal lädt IFF__Filter das Bild »Pic« von der Festplatte DH0: und wendet den ersten Relieffilter und den zweiten Gradientenfilter darauf an. Dann speichert das Programm das Bild in der RAM-Disk unter dem Namen »Bild«. Während des Vorgangs werden Informationen zum Programmlauf ausgegeben.

Mit dem Parameter »-OWN« sind Sie in der Lage, eigene Filter zu definieren. Dem Einfallsreichtum sind hier keine Grenzen gesetzt. Wählt man diese Option, fragt IFF__Filter zuerst nach der Größe der Filtermatrix. Diese muß ungerade sein, damit ein zentraler Punkt eindeutig festgelegt ist. Dies ist aber keine Einschränkung, denn man kann eine 2x2-Matrix in eine 3x3-Matrix einbetten, indem man die restlichen Zeilen und Spalten mit 0 füllt. Mit Ausnahme der Nullmatrix (alle Elemente sind 0) sind alle Matrizen erlaubt. In der Regel erzielen die Matrizen die besten Ergebnisse, die ausgewogen sind, das heißt die Summe aller Elemente ist Null. Für den eigenen Filter gilt, wie so oft, das Sprichwort: »Probieren geht über Studieren«. *René Beaupoil/kn*

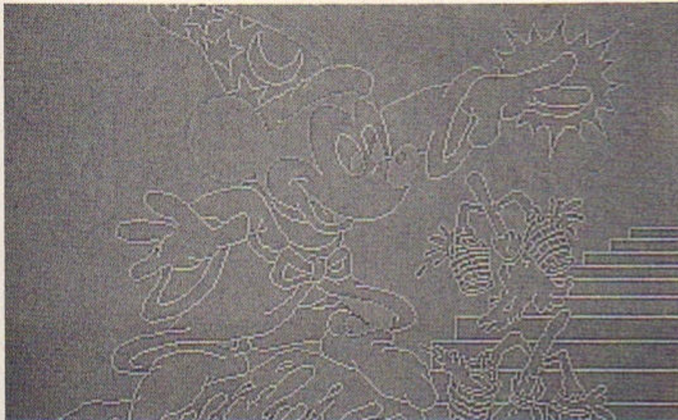


Bild 4. Das Originalbild (siehe Bild 1) bearbeitet mit dem Gradientenfilter 3 und Aufhellung (Faktor 4)

Filtermatrix	Bedeutung
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	**Tiefpaß:** Breite und Höhe der Matrix variabel (hier 7x3); Matrix mit 1 gefüllt. Unschärfe und Störpunkte verschwinden
+1 +4 +6 +4 +1 +2 +8 +12 +8 +2 0 0 0 0 0 -2 -8 -12 -8 -2 -1 -2 -6 -4 -1	**Relieffilter 1** (vertikal): Bild wie aus Stein gemeißelt. Um den Relieffilter horizontal zu bekommen, muß die Matrix um 90 Grad gedreht werden.
-1 -3 -3 -1 0 -3 -8 -6 0 +1 -3 -6 0 +6 +3 -1 0 +6 +8 +3 0 +1 +3 +3 +1	**Relieffilter 2** (diagonal): wie bei Relieffilter 1, nur etwas anderer Blickwinkel
-1 0 0 +1	**Gradientenfilter 1 diagonal:** Hervorheben der Kanten in diagonaler Richtung
-1 +1	**Gradientenfilter 2 vertikal:** Hervorheben der Kanten in vertikaler Richtung
+1 +1 +1 +1 -8 +1 +1 +1 +1	**Gradientenfilter 3 (alle 8 Richtungen):** Hervorheben der Kanten in allen 8 Richtungen.

Tabelle. So sehen die verschiedenen Filtermatrizen aus, die in optimierter Form berechnet werden

Programmname:	IFF__Filter
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.6
Aufrufe:	CC IFF__Filter -E250 +L LN IFF__Filter.o -LC32

```
Programmautor: Peter Sparlinek
------------------------------

ALO #include <stdio.h>
uZ  #include <exec/exec.h>
io  #include <libraries/dos.h>
hL  #define MIN(a,b) (a<b)?a:b
S2  #define ABS(x) (x>0)?(x):(-(x))
CY  #define INFO(s) if(info)puts(s)
7g  long *MyLock;
Zr  struct FileInfoBlock MyInfo;
2q  long adr;
uo  char *aktuell;
qb  FILE *in,*out,*fopen();
11  int cc,i,j,k,filtertyp,fnset=0,neg=0;
49  char *matrix,*erg,Zeile[100],string[60],text[5],c;
cU  long help,colors[32];
G8  unsigned char bmhd[25],jj,rot,blau,gruen;
LY  int helligkeit,width,anzbyte,height,ddepth,depth,f_anz,line
    s,info,count,pack,
uV4     planes,kompress=-10;
7MO char fo[]={0x46,0x4f,0x52,0x4d,0x00,0x00,0x00,0x00,
HGB           0x49,0x4c,0x42,0x4d};
aXO char cm[]={0x43,0x4d,0x41,0x50,0x00,0x00,0x00,0x30,
FAB           0x00,0x00,0x00,0x10,0x10,0x10,0x20,0x20,0x20,
OM            0x30,0x30,0x30,0x40,0x40,0x40,0x50,0x50,0x50,
1Y            0x60,0x60,0x60,0x70,0x70,0x70,0x80,0x80,0x80,
cc            0x90,0x90,0x90,0xa0,0xa0,0xa0,0xb0,0xb0,0xb0,
vF            0xc0,0xc0,0xc0,0xd0,0xd0,0xd0,0xe0,0xe0,0xe0,
MR            0xf0,0xf0,0xf0};
tJ0 /******************************** main ********************
    ************/
Ws  main(argc,argv)
tT  int argc;
V3  char **argv;
Ru  {
QK1  int c;
Qc   long zw;
I1   char s[20];
Pf   helligkeit=1;filtertyp=0;text[4]=0,pack=1,info=0;
AN   if(argc<2 || !strcmp(argv[1],"?") || i>99) Hilfe();
o3   for(i=2;i<argc;i++)
Y1   {
Hm2   strcpy(s,argv[i]);
d8    if(!strncmp(s,"-D",2) || !strncmp(s,"-d",2))
b4    {
Gr3    fnset=1;
oW     strcpy(string,&s[2]);
iD2   }
Vn    for(k=0;k<strlen(s);k++)s[k]=toupper(s[k]);
UZ    if(!strcmp(s,"-N"))neg=1;
pH    else if(!strcmp(s,"+I"))info=1;
1k    else if(!strcmp(s,"-P"))pack=0;
xv    else if(!strncmp(s,"-H",2))
kD    {
CI3    if(strlen(s)==3)
92     if(s[2]>'0' && s[2]<'5') helligkeit=s[2]-'0';
rM2   }
sN1  }
x2   ReadIFF(argv[1]); /* Liest IFF-Bild ein */
J3   setmem(erg,width*height,0x00);
rZ   Col_to_grey(); /* Umwandlung in Grautöne */
90   for(i=2;i<argc;i++)
tM   {
c72   strcpy(s,argv[i]);
13    for(k=0;k<strlen(s);k++)s[k]=toupper(s[k]);
3y    if(!strcmp(s,"-ID"))Id();
GA    else if(!strcmp(s,"-REL1"))Relief1();
RN    else if(!strcmp(s,"-REL2"))Relief2();
```

Listing. Bildverarbeitung mit »IFF__Filter«. Bitte mit dem Checksummer (Seite 143) eingeben.


```
  else if(!strcmp(s,"-GRA1"))Ableitung1();
  else if(!strcmp(s,"-GRA2"))Ableitung2();
  else if(!strcmp(s,"-GRA3"))Ableitung3();
  else if(!strcmp(s,"-LOW" ))Tiefpass();
  else if(!strcmp(s,"-OWN" ))Own();
 }
 if(!fnset)
 {
  strcpy(string,argv[1]);
  strcat(string,".Filter");
  if(neg) strcat(string,"n");
 }
 if(!(out=fopen(string,"w"))) exit(puts("Kann Zieldatei nic
ht öffnen!\n"));
 printf("Bild in Datei %s\n",string);
 PutIFF();
 fclose(out);
 FreeMem(matrix,width*height);
 FreeMem(erg,width*height);
}
/**********************   Funktionen   ******************
***********/
Hilfe()            /* Hilfstext bei falschen Aufrufparamete
rn */
{
 puts("USAGE IFF_Filter <filename> [Parameter...]");
 puts("Parameter:");
 puts("-d<filename>   Festlegen der Ausgabedatei");
 puts("-n             negativ");
 puts("-p             IFF-Bild ungepackt abspeichern");
 puts("+i             Info über Programmlauf");
 puts("-h<n>           Aufhellen des Bildes 0<n<5");
 puts("               1: normal ... 4 sehr hell");
 puts("-<typ>          Auswahl des Filters");
 puts("\tmit <typ>:");
 puts("\t\tID\tIdentität");
 puts("\t\tREL1\tRelief-Filter 1");
 puts("\t\tREL2\tRelief-Filter 2");
 puts("\t\tGRA1\tAbleitung diagonal");
 puts("\t\tGRA2\tAbleitung waag- und senkrecht");
 puts("\t\tGRA3\tAbleitung in alle 8 Richtungen");
 puts("\t\tLOW\tTiefpass");
 puts("\t\tOWN\teigener Filter");
 exit(1000);
}
Id()      /* Bild nur in Grauwerte umwandeln - kein Filter
*/
{
 register int x;
 INFO("Bild in Grauwerte - kein Filter");
 if(neg)
 {
  for(x=0;x<width*height;x++)
   *(erg+x)=14- *(matrix+x);
 }
 else
  for(x=0;x<width*height;x++)
   *(erg+x)= *(matrix+x);
}
Relief1()      /* Relieffilter vertikal */
{
 long zwischen,dw=2*width;
 int k,l,i1,i2,i3,i4,i5;
 INFO("RELIEF-FILTER 1");
 for(k=2;k<width-2;k++)
 {
  help=dw+k;
  i1=matrix[k-2]+4*matrix[k-1]+6*matrix[k]+4*matrix[k+1]+ma
trix[k+2];
  i2=matrix[help-width-2]+4*matrix[help-width-1]+6*matrix[h
elp-width]+4*matrix[help-width+1]+matrix[help-width+2];
  i3=matrix[help-2]+4*matrix[help-1]+6*matrix[help]+4*matri
x[help+1]+matrix[help+2];
  i4=matrix[help+width-2]+4*matrix[help+width-1]+6*matrix[h
elp+width]+4*matrix[help+width+1]+matrix[help+width+2];
  i5=matrix[help+dw-2]+4*matrix[help+dw-1]+6*matrix[help+dw
]+4*matrix[help+dw+1]+matrix[help+dw+2];
  for(l=2;l<height-2;l++)
  {
   zwischen=-i1-i2-i2+i4+i4+i5;
   if(neg)
    zwischen=7-(zwischen*7)/672;
   else
    zwischen=7+(zwischen*7)/672;
   erg[k+width*l]=(char)zwischen;
   i1=i2;i2=i3;i3=i4;i4=i5;
   help=k+width*(l+3);
   i5=matrix[help-2]+4*matrix[help-1]+6*matrix[help]+4*matr
ix[help+1]+matrix[help+2];
  }
 }
}/* Relief 1 */
Relief2()      /* Relieffilter diagonal */
{
 int k,l,m,dw=width*2;
 INFO("RELIEF-FILTER 2 (diagonal)");
 for(k=2;k<width-2;k++)
 {
  for(l=2;l<height-2;l++)
  {
   help=k+l*width;
   m=-matrix[help-dw-2]-3*matrix[help-dw-1]-3*matrix[help-d
w]-matrix[help-dw+1]
    -3*matrix[help-width-2]-8*matrix[help-width-1]-6*matrix
    [help-width]+matrix[help-width+2];
   m+=-3*matrix[help-2]-6*matrix[help-1]+6*matrix[help+1]+3
   *matrix[help+2]
    -matrix[help+width-2]+6*matrix[help+width]+8*matrix[hel
    p+width+1]+3*matrix[help+width+2];
   m+=matrix[help+dw-1]+3*matrix[help+dw]+3*matrix[help+dw+
   1]+matrix[help+dw+2];
   if(neg)
    m=7-(m*7)/490;
   else
    m=7+(m*7)/490;
   erg[k+width*l]=(char)m;
  }
 }
}
Ableitung1()
{
 int k,l,m;
 INFO("GRADIENTENFILTER 1 (diagonal)");
 for(k=1;k<width;k++)
 {
  for(l=1;l<height;l++)
  {
   help=k+l*width;
   m=matrix[help]-matrix[help-width-1];
   if(neg)
    m=7-(m*7)/14;
   else
    m=7+(m*7)/14;
   erg[k+width*l]=(char)m;
  }
 }
}
Ableitung2()
{
 int k,l,m;
 INFO("GRADIENTENFILTER 2 (vertikal)");
 for(k=0;k<width;k++)
 {
  for(l=1;l<height;l++)
  {
   help=k+l*width;
   m=matrix[help]-matrix[help-width];
   if(neg)
    m=7-(m*7)/56;
   else
    m=7+(m*7)/56;
   erg[k+width*l]=(char)m;
  }
 }
}
Ableitung3()
{
 int k,l,m;
 INFO("GRADIENTENFILTER 3 (alle 8 Richtungen)");
 for(k=1;k<width;k++)
 {
  for(l=1;l<height;l++)
```

```
  {
   help=k+l*width;
   m=matrix[help-width-1]+matrix[help-width+1]+matrix[help+
   width+1]
    +matrix[help+width-1]+matrix[help-width]+matrix[help+1]
    +matrix[help+width]+matrix[help-1]-8*matrix[help];
   if(neg)
    m=7-(m*7)/112;
   else
    m=7+(m*7)/112;
   erg[k+width*l]=(char)m;
  }
 }
}
Tiefpass()
{
 int i,j,z,x,y,xl,yl,xdiv,ydiv,ywidth,xy,hh,ok;
 char c;
 puts("TIEFPASS");
 for(ok=0;ok==0;)
 {
  puts("\nBitte Größe des Filters eingeben. (Nur ungerade W
  erte)");
  printf("waagrecht: (3<= x <=99) "); scanf("%d",&xl);
  printf("senkrecht: (3<= y <=17) "); scanf("%d",&yl);
  if(xl>2 && yl>2 && xl<100 && yl<18 && (xl%2) && (yl%2
  )) ok=1;
 }
 if(!fnset)
  sprintf(string,"%s_%d_%d",string,xl,yl);
 xdiv=xl/2;ydiv=yl/2;
 xy=xl*yl;ywidth=ydiv*width;
 for(x=ywidth;x<width+ywidth;x++)
  for(y=x-ywidth;y<=x+ywidth;y+=width)
   erg[x]=(char)(erg[x]+matrix[y]);
 for(x=0;x<width;x++)
  for(y=ywidth+width+x;y<(height-ydiv)*width+x;y+=width)
   erg[y]=(char)(erg[y-width]-matrix[y-width-ywidth]+matrix
   [y+ywidth]);
 movmem(erg,matrix,width*height);
 setmem(erg,width*height,0x00);
 for(y=ywidth;y<(height-ydiv)*width;y+=width)
 {
  for(hh=x=0;x<xl;x++) hh+=matrix[x+y];
  for(x=xdiv;x<width-xdiv;x++)
  {
   erg[x+y]=(char)(hh/xy);
   hh=(hh+matrix[x+y+xdiv+1])-matrix[x+y-xdiv];
  }
 }
}
Own()      /* Eigener Filter */
{
 int *fil,xm=0,ym=0,xo,yo,zw,ii,jj,neg,pos,kk,ll,xf,i1,i2;
restart:
 while(!(xm>0 && xm<width/4) && !(ym>0 && ym<height/4))
 {
  printf("Bitte Filtergröße eingeben\nwaagrecht (0<x<%d)
  ",width/4);
  scanf("%d",&xm);
  printf("senkrecht (0<y<%d) ",height/4);
  scanf("%d",&ym);
 }
 if(!(xm%2)) printf("x auf %d angehoben\n",++xm);
 if(!(ym%2)) printf("y auf %d angehoben\n",++ym);
 if(!(fil=(int *)malloc(sizeof(int)*xm*ym)))
 {
  puts("Kann nicht genügend Speicher belegen\nVersuchen Sie
   kleinere Werte");
  goto restart;
 }
 pos=neg=0;
 puts("Eingabe der Filtermatrix");
 for(ii=0;ii<ym;ii++)
  for(jj=0;jj<xm;jj++)
  {
   printf("\nFilter[%d;%d]=",ii+1,jj+1); scanf("%d",&zw);
   if(zw>0) pos+=zw; else neg-=zw;
   fil[ii*xm+jj]=zw;
  }
 if(pos==neg &&  neg==0)
 {
  puts("Nullmatrix nicht erlaubt !");
  goto restart;
 }
 puts("Filter ok ? (j|y|n)");
 {char c;
  for(c=' ';c!='y' && c!='j' && c!='n';c=getchar());
  if(c=='n') goto restart;
 }
 INFO("Berechnung des gefilterten Bildes. Es kann etwas dau
 ern ....");
 pos+=neg;
 for(ii=xm/2;ii<width-xm/2;ii++)
 {
  for(jj=width*(ym/2)+ii;jj<width*(height-ym/2)+ii;jj+=wid
  th)
  {
   zw=0;i1=xm/2+width*(ym/2);i2=width*(ym/2)-xm/2;
   for(kk=jj-i1,xf=0;kk<=jj+i2;kk+=width)
    for(ll=kk;ll<xm+kk;ll++,xf++)
     zw+= *(fil+xf) * matrix[ll];
   erg[jj]=(char)((zw+neg*14)/pos);
  }
 }
 free(fil);
}
Col_to_grey()      /* Grauwertumwandlung */
{
 register int ii,jj,kk;
 INFO("GRAUWERTUMWANDLUNG");
 kk=width*height;
 for(jj=0;jj<kk;jj++)
  matrix[jj]=(char)colors[matrix[jj]];
}
ReadIFF(bildname)   /* Bild im IFF-Format einlesen */
char *bildname;
{
 int body_gelesen=0,loop,lw,laenge,breite;
 long lang;
 MyLock=Lock(bildname,ACCESS_READ);
 if(!MyLock) exit(printf("Illegal Filename %s\n",bildname))
 ;
 if(!Examine(MyLock,&MyInfo)) exit(printf("Can't get %s\n",
 bildname));
 laenge=MyInfo.fib_Size+1;
 UnLock(MyLock);
 aktuell=adr=AllocMem(laenge,MEMF_CHIP);
 in=Open(bildname,MODE_OLDFILE);
 Read(in,adr,laenge);
 Close(in);
 INFO("Lesen IFF-Bild");
 while(!body_gelesen)
 {
  MyRead(text,4);
  if(!strncmp(text,"BMHD",4))
  {
   MyRead(bmhd,24);  /* BitMapHeader einlesen */
   breite=bmhd[4]*256+bmhd[5];
   if(breite%16)width=((breite>>4)+1)<<4;
   else width=breite;
   height=bmhd[6]*256+bmhd[7];
   depth=(char)bmhd[12];
   kompress=(char)bmhd[14];
   ddepth=1<<depth;  /* ddepth =2^depth */
   if(!(matrix=AllocMem(width*height,MEMF_CLEAR)))
    exit(puts("Kann nicht genügend Speicher belegen!"));
   if(!(erg=AllocMem(width*height,MEMF_CLEAR)))
    {
     FreeMem(matrix,width*height);
     exit(puts("Kann nicht genügend Speicher belegen!"));
    }
  if(info)
   {
    puts("Bilddaten:");
    printf("-Breite    :%d\n-Hoehe     :%d\n-Bitplanes :%d
    \n-Farben    :%d\n-gepackt   :",breite,height,depth,dde
    pth);
```

Listing. Bildverarbeitung mit »IFF__Filter« (Fortsetzung)

```
      if(kompress) puts("Ja"); else puts("Nein");
    }
  } /* BMHD */
  else if(!strncmp(text,"CMAP",4))
  {
    INFO("Farbtabelle (CMAP)");
    MyRead(&f_anz,4);
    f_anz/=3;
    for(loop=0;loop<f_anz;loop++)
    {
      MyRead(&rot,1);
      MyRead(&gruen,1);
      MyRead(&blau,1);
      cc=(char)((((rot>>4)+(gruen>>4)+(blau>>4))*
          (14+!filtertyp)*helligkeit)/45);
      colors[loop]=(char)(MIN(cc,0x0e+!filtertyp));
    }
  } /* CMAP */
  else if(!strncmp(text,"BODY",4))
  {
    INFO("Bilddaten (BODY)");
    if(kompress<0) exit(puts("Kein BMHD-Chunk!"));
    MyRead(text,4); /* CHUNK-Länge überlesen */
    if(!kompress)
    {
      INFO("einlesen\n");
      for(lines=0;lines<height;lines++)
      {
        lw=lines*width;
        for(planes=1;planes<ddepth;planes<<=1)
        {
          MyRead(erg,width/8);
          for(j=i=0;i<width/8;i++)
            for(k=128;k>0;j++,k>>=1)
              if(erg[i] & k) matrix[lw+j]+=planes;
        }
      }
    }
    else
    {
      INFO("Entpacken");
      for(lines=0;lines<height;lines++)
      {
        lw=lines*width;
        for(planes=1;planes<ddepth;planes<<=1)
        {
          i=0;
          while(i<width/8)
          {
            MyRead(&jj,1);
            if(jj>0 && jj<128)
            {
              MyRead(&erg[i],jj+1);
              i+=jj+1;
            }
            else
            {
              jj=(char)(-jj+1);
              MyRead(&erg[i],1);
              for(k=1;k<jj;k++) erg[i+k]=erg[i];
              i+=jj;
            }
          }
          for(i=j=0;i<width/8;i++)
            for(k=128;k>0;j++,k>>=1)
              if(erg[i] & k) matrix[lw+j]+=planes;
        }
      }
    }
    body_gelesen=1;
  } /* BODY */
  else if(!strncmp(text,"FORM",4))
  {
    MyRead(&laenge,4);
  }
  else if(!strncmp(text,"ILBM",4))
  {}
  else
  {
    MyRead(&lang,4);
    for(k=0;k<lang;k++)MyRead(laenge,1);
  }
 } /* while */
 FreeMem(adr,laenge);
} /* ReadIFF */
PutIFF()      /* Bild im IFF-Format abspeichern */
{
 int lw;
 INFO("Schreiben des IFF-Bildes");
 for(lines=0;lines<height;lines++)
 {
  lw=lines*width;
  for(planes=1;planes<16;planes<<=1)
  {
   cc=0;
   for(i=0;i<=width;i++)
   {
    cc<<=1;
    if(erg[lw+i] & planes) cc++;
    if(!(i%8) && i)
    {
     if(pack)
      Zeile[i/8-1]=(char)cc;
     else
      matrix[count++]=(char)cc;
     cc=0;
    }
   }
   if(pack) Packen();
  }
 }
 fo[6]=(char)((4+28+56+8+count+count%2)/256);
 fo[7]=(char)((4+28+56+8+count+count%2)%256);
 fwrite(fo,12,1,out);  /* FORM-Chunk */
 fwrite("BMHD",4,1,out);
 bmhd[14]=(char)pack;
 bmhd[12]=(char)4;
 fwrite(bmhd,24,1,out);  /* BMHD-Chunk */
 fwrite(cm,56,1,out);    /* CMAP-Chunk */
 fwrite("BODY",4,1,out);
 fputc(0x00,out);fputc(0x00,out);
 fputc((char)((count+count%2)/256),out);
 fputc((char)((count+count%2)%256),out);
 fwrite(matrix,count,1,out);
 if(count%2) fputc(0x00,out); /* auf gerade Anzahl bringen
 */
} /* PutIFF */
Packen()      /* IFF-Bild packen */
{
 int last,i,j;
 char m;
 last=width/8;
 i=0;
 while(i<last)
 {
  m=Zeile[i];
  for(j=i+1;m==Zeile[j] && j<last;j++);
  if(j!=i+1)
  {  /* Jetzt wird gepackt */
   matrix[count++]=(char)(1-j+i);
   matrix[count++]=m;
   i=j;
  }
  else
  {  /* Hier wird ungepackt übernommen */
   for(;m!=Zeile[j]&&j<=last;m=Zeile[j++]);
   if(j<last) j-=2;
   j=MIN(last-1,j);
   matrix[count++]=(char)(j-i);
   for(;i<=j;matrix[count++]=Zeile[i++]);
  }
 } /* while */
} /* Packen */
MyRead(nach,anz)   /* Hilfsfunktion für ReadIFF() */
char *nach;
int anz;
{
 int i;
 for(i=0;i<anz;i++) *nach++=*aktuell++;
}
(C) 1989 M&T
```

Listing. Bildverarbeitung mit »IFF_Filter« (Schluß)

Grafik in der dritten Dimension

»Ray Tracer« heißt wörtlich übersetzt »Strahlenverfolger«. Wer aber glaubt, mit dem folgenden Programm könne er seinen Amiga als Geigerzähler einsetzen, der irrt sich. Die Strahlen sind in unserem Fall harmloses Licht und verfolgt werden sie zur dreidimensionalen Darstellung.



AMIGA SONDERHEFTE

1 Rund um den Amiga

Ausführliche Kurse, Bauanleitung, Tips&Tricks

Das Sonderheft 1 enthält einen ausführlichen Basic-Kurs - alles über den richtigen Umgang der alternativen Benutzeroberfläche "Command Interface Line". Bauanleitung: Mit dem Digitizer und der zugehörigen Software "Digisoft Plus" tauchen Sie ein in die Welt der digitalen Klänge. Tips&Tricks: aufgeteilt nach vier Bereichen: Einsteiger, Insider, Amiga-Basic und CLI.





2 Listings der Spitzenklasse

Tolle Anwendungen, Tips, Tricks und Tools

"Fontdesigner", "Objekt-Editor", "Haushaltsbuch", "Keyboard-Master", "Fast Load-Copy", "Super-Copy" sind nur einige von vielen Anwendungen im Sonderheft 2. Außerdem: Ratschläge zu Superbase und zwei außergewöhnliche Spiele.

3 C und Assembler

Zwei Super-Kurse und viele Tips und Tricks

Der große C-Kurs für alle Umsteiger. Programmieren Sie ab sofort mit kräftiger Unterstützung durch Betriebssystem und Libraries. Der Assembler-Kurs enthüllt alle Geheimnisse des 68000er Prozessors.





4 Basic und Spiele

10 Spitzenspiele zum Abtippen in AMIGA-Basic

Die absoluten Spiele Top-Hits für den AMIGA in einer Übersicht. Mit Klassikern wie "Shanghai" und vielen brandneuen Hits. Ein ausführlicher Basic-Kurs führt zu Programmen mit rasanten Geschwindigkeiten.

5 Public Domain

Software zum Nulltarif: Das Beste aus aller Welt

Die ganze Faszination der Grafik in einem Projekt. Wichtigster Themenbereich werden Utilities mit zahlreichen Hilfsprogrammen sein.





7 Spiele-programmierung

Knowhow, Projekte, Spiele

Knowhow: Wir lüften die Geheimnisse von Blitter, Copper & Co. **Projekte:** Die erste 3D-Vektor Library für den Amiga ermöglicht völlig neue Erlebnisse in der Grafik-Welt. Lesen Sie über die Entstehung neuer Spiele von der Idee bis zum Programm. **Spiele:** Fünf brandheiße Spiele aus diesem Heft sind - neben vielen Tools und Utilities - auf Diskette erhältlich.



6 Anwendungen

Workshops: Experten verraten alle Tricks im Umgang mit den professionellen Programmen Sonix, Superbase und Beckertext. **Überblick:** Anwendungsprogramme und die Vorstellung der besten Programme. **Top-Programme fast umsonst:** Hochkarätige Anwendungen zur Textverarbeitung, Kontenführung und Dateiverwaltung bestechen durch ausgereifte Benutzerführung.

Ich bestelle: ____ Ausgaben AMIGA SH Nr.____
____ " " " Nr.____
____ " " " Nr.____
____ " " " Nr.____
____ " " " Nr.____

Insgesamt: ____ Ausgaben für 16,-DM pro Exemplar

Name, Vorname ____________________

____

Straße, Hausnummer ____________________

PLZ, Wohnort ____________________

Rechnungssumme ________ DM

zzgl. 3,-DM Versandkosten ________ 3,-DM

Rechnungsbetrag insgesamt ________ DM

Schicken Sie bitte die ausgefüllte Karte zusammen mit dem Rechnungsbetrag als Verrechnungsscheck im Briefumschlag an: Markt&Technik Verlag AG, Leserservice, Hans-Pinsel-Str.2 in 8013 Haar bei München

AMIGA BESTELLCOUPON

Ray Tracing« nennt man die perspektivische Darstellung von dreidimensionalen Objekten unter Berücksichtigung der Lichtstrahlen. Was in diesem kurzen Satz an List und Tücke für den Programmierer steckt, wird erst vollends klar, wenn Sie sich verdeutlichen, was dabei berücksichtigt werden muß.

So müssen nicht nur die Abmessungen der Objekte (in unserem Fall Kugeln) der Perspektive entsprechend umgerechnet werden. Es muß auch der Lichteinfall berücksichtigt, die Spiegelungen richtig dargestellt (möglicherweise auf einer gekrümmten Oberfläche) und eventuell einzelne Flächen durchsichtig gemacht werden.

Verkürzte Rechenzeiten

Bei weniger leistungsfähigen Computern erweisen sich derartige Programme oft genug als wahre »Rechenzeitkiller«, die den Prozessor für erhebliche Zeit lahmlegen. Aber auch der Amiga – mit 68000er und dem Customchip Blitter sehr gut gerüstet – ist mit dieser Aufgabe stark gefordert.

So kann die Berechnung einer Grafik mit mehreren Elementen im Interlace-Modus durchaus zwei bis drei Stunden in Anspruch nehmen. Das heißt aber nicht, daß Sie sich einen zweiten Amiga kaufen müssen, weil Ihrer mit Raytracing demnächst ausgelastet ist. Es gibt einige Möglichkeiten, die Ausgabe zu beschleunigen oder notfalls abzubrechen.

Solange Sie an Ihrer Grafik noch Änderungen vornehmen, sollten Sie grundsätzlich im »Low-Resolution«-Modus arbeiten. Dieser ist etwa viermal so schnell wie der Interlace-Modus. Eine erhebliche Verkürzung der Rechenzeit bietet die Berechnung im »Test«-Modus. Diese stellt die Grafik im Briefmarkenformat dar. Sie brauchen auf eine Grafik nur wenige Minuten warten, für die sonst einige Stunden nötig wären.

Wollen Sie weitere Rechenzeit einsparen, ziehen Sie einfach während des Bildschirmaufbaus den Screen nach unten (oder stellen im Menü »verdeckt« ein). Sie müssen allerdings darauf achten, den Screen wieder ganz nach oben zu ziehen, bevor Sie mit »Window ändern« die Einstellungen bearbeiten. Denn dieser Menüpunkt öffnet ein eigenes Fenster, das sich nur durch ein Gadget am unteren Rand beseitigen läßt. Sollten Sie Ihren Amiga zwischendurch für dringende andere Aufgaben benötigen, so können Sie die Berechnung mit einem einfachen Mausklick an einer beliebigen Stelle des Screens abbrechen. Speichern Sie jetzt den derzeitigen Stand der Grafik. Wenn Sie weitermachen wollen, laden Sie diese und arbeiten an derselben Stelle weiter. Leider ist es uns aufgrund der Länge dieses Programms nicht möglich, das Listing abzudrucken, da sonst kaum Platz für weitere Programme in diesem Heft übrigbliebe.

Der Raytracer ist nur auf unserer Programmservice-Diskette enthalten, die Sie mit der Karte auf Seite 34 bestellen können.

Programm-Service

Starten Sie den Raytracer entweder vom CLI mit »rt« und <RETURN> oder von der Workbench per Doppelklick auf das Icon. Sie sehen jetzt einen neuen Screen mit der Titelleiste. Drücken Sie die rechte Maustaste, dann sehen Sie vier Menüs: »RayTracer«, »Auflösung«, »Berechnen« und »Extras«:

Auflösung: Wählen Sie den Punkt »High« an, so wird der Bildschirm auf »High-Resolution« und Interlace umgestellt (640 x 512 Punkte).

Bevor Ihnen vom Flackern die Augen schmerzen, schalten Sie lieber zurück (mit dem Menüpunkt »Low«). Der Bildschirm sollte jetzt wieder die normale Auflösung von 320 x 256 Punkten haben.

Berechnen: Hier finden Sie die beiden Punkte »Test« und »Start«. Bitte zügeln Sie vorerst Ihre Neugier und lassen Sie den Punkt »Start« links liegen. Dieser bringt längere Wartezeiten mit sich. Starten Sie statt dessen einen »Test«-Lauf. Dieser berechnet – weil noch keine Objekte definiert wurden – in wenigen Sekunden die Grafik. Sie sehen jetzt in der linken oberen Ecke eine Grafik im Briefmarkenformat, die ein rotgrünes Schachbrettmuster und einen schmalen blauen Streifen als Horizont anzeigt.

Natürlich ist es nicht Sinn eines Raytracers, eine Flachlandsilhouette mit Schachbrettmuster auf den Bildschirm zu zaubern. Dafür gibt es weit anspruchslosere Programme.

Extras: Gehen wir also daran, das erste Objekt zu definieren. Zu diesem Zweck wählen Sie im Menü »Extras« den Punkt »W. ändern«. Sie sehen ein neues Fenster (Bild 1), in dem Sie alle wichtigen Parameter einstellen.

```
-*- Ray Tracer -*-
Bildschirmmitte X = 0.000000
                Y = 0.000000
                Z = 17.999999
Blickrichtung   X = 5.000000
                Y = 10.000000
                Z = -2.600000
Bildschirmabstand = 0.280000
Brechungszahl     = 2.560000
                    »
Kugelnummer [RET] = 0
                    «
Durchmesser       = 50
Kugelposition   X = -50
                Y = 200
                Z = 200
Sp/Gl/Li  (1/2/3) = 3
Kugel löschen (!)   ø
            OK
```

Bild 1. Mit dem Menüpunkt »W. ändern« stellen Sie alle wichtigen Daten für Ihre Grafik ein

Sie sehen an der aktuellen Einstellung, daß ein Objekt bereits definiert ist. Es handelt sich dabei um die Kugel mit der Nummer Null, die als Lichtquelle dient. Sie können die Position dieser Lichtquelle ändern, indem Sie mit der Maus zweimal auf die entsprechenden Werte klicken und diese editieren, sobald der Cursor auftaucht.

Ähnlich gehen Sie vor, wenn Sie eine neue Kugel plazieren wollen. Klicken Sie zweimal auf die Nummer der Kugel (im Moment »0«) und tragen Sie dort eine Eins ein. Diese Änderung wird allerdings erst akzeptiert, wenn Sie mit einem Klick auf das Feld »[RET]« bestätigen. Sie sehen jetzt die aktuellen Maße Ihrer neuen Kugel angezeigt, nämlich fünf Nullen. Geben Sie ihr neue Werte, indem Sie die alten wie oben beschrieben ändern. Achten Sie darauf, daß Sie am Anfang Werte eintragen, die im darstellbaren Bereich liegen – später können Sie immer noch experimentieren.

Wir schlagen vor, daß Sie folgende Startwerte eintragen: Durchmesser 50, X-Position 50, Y-Position 100, Z-Position 200 und bei »Sp/Gl/Li« eine 2. Der letzte Wert unterscheidet zwischen spiegelnder (1) und glatter (2) Oberfläche beziehungsweise einer Lichtquelle (3).

Die erste eigene »Landschaft«

Klicken Sie mit der Maus jetzt auf »[RET]«, um die neuen Einstellungen zu bestätigen und danach auf das »ok«-Gadget am unteren Bildschirmrand.

Wenn Sie jetzt noch einmal »Test« anwählen, sehen Sie die Grafik mit Ihrer neu definierten Kugel am rechten Rand. Außerdem dürfte Ihnen aufgefallen sein, daß die Berechnung der Grafik jetzt deutlich mehr Zeit in Anspruch nimmt.

Um einen ersten Eindruck zu erhalten, sollten Sie Ihren Amiga vielleicht für zwei bis drei Stunden »beschäftigen«. Stellen Sie den Interlace-Modus ein (mit »Auflösung«/»High«), starten Sie die Berechnung (mit »Berechnen«/»Start«) und ziehen Sie den Screen ganz nach unten. Der Lohn des Wartens ist eine Grafik, die zwar leicht flackert (wegen Interlace), aber immerhin einen kleinen Eindruck von der Leistungsfähigkeit des Amiga vermittelt. Wenn Sie alles genau befolgt haben, sollte das Bild etwa so aussehen, wie das Foto im Inhaltsverzeichnis, das bei unseren ersten Versuchen entstand.

Wenn Sie nach diesem Vorgeschmack Lust zum Experimentieren bekommen haben, steht dem nichts mehr im Wege. Sehen Sie sich den Menüpunkt »W. ändern« etwas genauer an. Sie finden unter anderem Möglichkeiten, Ihren Blickwinkel (unter »Blickrichtung«), den Abstand vom Ort des Geschehens (mit »Bildschirmabstand«) und die Brechungszahl zu verändern. Experimentieren Sie mit diesen Werten. Wenn Sie eine Einstel-

lung gefunden haben, die Sie vollkommen zufriedenstellt, können Sie diese mit »E. sichern« speichern. Das Programm legt eine Datei »RTWerte« an, aus der beim nächsten Start die Voreinstellungen geholt werden.

RayTracer: Auf ähnliche Art können Sie fertige Grafiken speichern und wieder laden. Die entsprechenden Menüpunkte finden Sie unter »Raytracer« (»Laden« und »Saven«). Beide verzweigen in einen Filerequester, der nur Directories oder selbstgespeicherte Grafiken erkennt. Dabei wird hinter jedem von Ihnen gewählten Namen entweder »-l« (wenn es sich um eine Low-Resolution-Grafik handelt) oder »-h« angehängt. Diese Erweiterung können Sie beim Speichern ebenso wie beim Laden weglassen.

Mit dem dritten Menüpunkt (»Ende«) sollten Sie vorsichtig umgehen. Er respektiert Ihren Wunsch und reagiert prompt, ohne vorher zu fragen, ob Sie nicht noch speichern wollen. Neben den bereits besprochenen Menüpunkten können Sie im »Extras«-Menü die Titelleiste ein- und ausblenden, ein Countdown-Fenster einrichten, das Sie informiert, wie viele Zeilen noch dargestellt werden müssen, und den aktuellen Screen sichtbar beziehungsweise unsichtbar machen. Auch diese Einstellungen werden in RTWerte gespeichert, so daß sie nicht jedesmal neu definiert werden müssen, sondern automatisch beim Start geladen werden. Das spart Zeit und erleichtert die Arbeit.

Mit den Informationen aus diesem Artikel und etwas Experimentierfreude sollten Sie in der Lage sein, fantastische Grafiken auf den Bildschirm zu zaubern. Wenn Sie mehr über die theoretischen Grundlagen dieses Themas wissen wollen, empfehlen das Buch »3-D-Grafik-Programmierung« von Data Becker, in dem Sie viele interessante Informationen zu diesem Gebiet finden. *Arno Gölzer/kn*

3-D-Grafik-Programmierung: Jennrich, Massmann, Schulz; Data Becker; 284 Seiten; ISBN 3-89011-174-2; Preis 59 Mark

---

Bewegung in 3 Dimensionen

Bewegte, perspektivische, dreidimensionale Gittergrafiken möchten Sie? »3DAnim« liefert sie Ihnen. Je nach Geschmack in Basic- oder Assemblerquellcode.

Von Harry Kellermeyer und René Beaupoil

Mit »3DAnim« können Sie dreidimensionale Objekte mit Perspektive auf dem Bildschirm darstellen. Für diesen Zweck stellt Ihnen das Basic-Programm eine eigene Sprache zur Verfügung. 3DAnim arbeitet wie ein Interpreter und generiert Quellcode in Basic oder Assembler. Die Geschwindigkeit in Basic ist erstaunlich. Lassen Sie sich überraschen.

Um dreidimensionale Bewegungen zu beschreiben, braucht man ein Koordinatensystem mit drei Achsen und einem Ursprung. Dieser befindet sich direkt auf dem Monitor links unten. Nach rechts wird die x-Achse, nach oben die y- und gleichsam »in den Monitor hinein« die z-Achse angetragen. Ausgegangen wird von einem Quader, dessen vordere Seite die Monitorfläche ist und dessen Tiefe genauso groß ist wie die Breite. Die Einheit ist in alle Richtungen ein Hires-Punkt. Dadurch treten keine Verzerrungen auf (640 x 512 x 640 virtuelle Punkte), obwohl die eigentliche Animation am Ende bei 640 x 256 Punkten abläuft. Natürlich können auch Punkte außerhalb dieses Raums positioniert werden, nur sind diese nicht unbedingt sichtbar. Um die Zeit noch mit ins Spiel zu bringen, geht das Programm so vor: Die Bewegung wird, wie beim Daumenkino, in Einzelbilder aufgelöst. Diese Bilder werden dann nacheinander dargestellt, wobei sich nach dem letzten Bild das Ganze wiederholt.

3DAnim versteht 25 Befehle, die zur Erzeugung der Bewegung nötig sind. Sie werden im folgenden erläutert. Parameter in eckigen Klammern sind optional, müssen also nicht angegeben werden.

■ DIM p,n[,opt]

Dieser Befehl muß der erste in der Liste sein und darf nur dort vorkommen. Es wird angegeben, wie viele Bilder man insgesamt benutzen will: DIM p,n bedeutet, daß auf n Bildern je p Punkte dargestellt werden sollen. Dabei geht die Numerierung von 0 bis p-1 und von 0 bis n-1. Wird der Parameter »opt« angegeben, werden keine Linien zum Fluchtpunkt gezeichnet (siehe auch PERS).

■ PERS x,y,z

Wenn Sie das Programm ausprobieren, stellen Sie fest, daß Ihr Computer außer Ihren Objekten noch zwei weitere Linien zeichnet. Es sind Orientierungslinien, die zum Fluchtpunkt führen, also zu dem Punkt, in dem alle zur z-Achse parallelen Linien scheinbar zusammenstoßen. Sie können den Fluchtpunkt durch diesen Befehl selbst bestimmen. Wenn Sie die beiden Linien nicht im Bild haben wollen, geben Sie beim DIM-Befehl einfach einen dritten Parameter an (DIM 4,60,0).

■ ORIG x,y,z

Der Ursprung des Koordinatensystems wird neu definiert. Die Angaben beziehen sich immer auf das ursprüngliche Koordinatensystem, das heißt zweimal derselbe ORIG-Befehl bewirkt dasselbe wie nur einer. Es werden auch die Koordinaten aller Punkte auf den neuen Nullpunkt umgerechnet.

■ RES x[,y[,z]]

Mit diesem Befehl kann man im Koordinatensystem eine andere Auflösung wählen. Dabei können auch Verzerrungen entstehen. Fehlt ein Parameter, so wird er der Pixelgröße in x-Richtung angepaßt. Also entspricht »RES 320« dem Befehl »RES 320,256,320«. Der Ursprung, Fluchtpunkt und sämtliche Punkte werden entsprechend umgerechnet, so daß sie auf dem Bildschirm an derselben Stelle erscheinen. Ist ein Parameter negativ, so kehrt sich die Richtung der Achse um.

■ INKS r1,g1,b1[,r0,g0,b0]

Die Parameter liegen im Bereich von 0 bis 15 und bestimmen die RGB-Werte der Farben 1 und 0. Wenn Sie die letzten drei Werte weglassen, werden sie automatisch 0, das heißt der Hintergrund ist schwarz. Voreinstellung ist »INKS 15,15,15, 0,0,0«, also weiße Objekte auf schwarzem Hintergrund.

■ MASK flagx,flagy,flagz

Auf diese Weise kann man erreichen, daß bei den Befehlen SET, ACC, DEC, SHI, SHII und COPY einzelne Komponenten ausmaskiert werden, wenn das entsprechende Flag 0 ist. Nach dem Kommando »MASK 1,1,0« werden durch einen COPY-Befehl nur noch die x- und y-Werte kopiert. Eine andere Anwendung ist, eine Rotation eines Körpers um eine zur x-Achse parallelen Achse mit entgegengesetzter Eigenrotation auf ein simples Auf und Ab zu reduzieren (durch Ausschalten der z-Komponente), um dies dann mittels »SHI dx,0,0« in einen Sinus zu verwandeln.

■ LET var,wert

Einer Variablen (v0 bis v100) kann ein Wert zugewiesen werden.

■ SET p,x,y,z[,a[,b]]

Damit man Punkte bewegen kann, muß erst ein Anfang gemacht werden. Dies geschieht durch den Befehl SET. Dem Punkt p werden von Bild »a« bis Bild »b« die Koordinaten (x, y, z) zugewiesen. Fehlt b, so ist b gleich a. Ist weder a noch b vorhanden, gilt die Zuweisung für alle Bilder. Folgen mehrere SET-Befehle hintereinander und wird jeweils der nächste Punkt definiert, können die Parameter, durch einen Doppel-

Harry Kellermeyer

Der Autor begann mit seinem Computer-Hobby Ende 1985 auf einem Sharp PC 1401. Wegen der beschränkten Fähigkeiten folgte bald ein Schneider CPC 464. Im September 1987 schaffte sich Harry Kellermeyer dann einen Amiga 500 an. »3DAnim« war ursprünglich ein Programm, das einen rotierenden Oktaeder auf dem CPC 464 darstellte. Mit der größeren Leistungsfähigkeit des Amiga wuchs das Programm dann immer weiter. Das Resultat ist »3DAnim«, das vor Ihnen liegt. Mit dem Gewinn von 2000 Mark (Programm des Monats im AMIGA-Magazin 8/89) finanziert der Autor seinen Urlaub.

punkt getrennt, aneinandergehängt werden. Der jeweils erste Parameter muß dann aber vom zweiten Punkt an weggelassen werden. So wird aus

```
SET 4,1,1,1
SET 5,,8
SET 6,8,3,3,2
```

schließlich

```
SET 4,1,1,1:,8:8,3,3,2
```

■ LINE punkteliste[:punkteliste[:punkteliste ...]]

Mit LINE wird festgelegt, in welcher Reihenfolge die Punkte jeweils miteinander verbunden werden sollen. Es werden zuerst die Punkte in der ersten Liste zusammenhängend gezeichnet, neu positioniert, weitergezeichnet und so weiter. Dabei besteht eine Punkteliste aus durch Kommata getrennten Parametern, die entweder ein Term oder ein Intervall (Term1..Term2) sind. So bedeuten beispielsweise:

5 Punkt 5
3..6 Punkte 3 bis 6
6..3 Punkte 6 bis 3 (in dieser Reihenfolge)
..10 Punkte 0 bis 10
4.. alle Punkte ab 4
.. alle Punkte

Kommen in einem Programm mehrere LINE-Anweisungen vor, so werden die einzelnen Punktelisten vor der Interpretation einfach aneinandergehängt und durch einen Doppelpunkt voneinander getrennt. Das heißt, die Befehle

```
LINE 2,1,3
LINE 4,5,6
```

bewirken dasselbe wie

```
LINE 2,1,3:4,5,6
```

■ REC Punkteliste:x0,y0,z0 ,dx,dy,dz

Den 4 Punkten in der Liste werden Koordinaten zugewiesen, sodaß ein Rechteck entsteht. Der erste Punkt ist bei (x0, y0, z0), die restlichen ergeben sich, indem man gegen den Uhrzeigersinn weitergeht. Das Rechteck liegt immer parallel zu einer Koordinatenebene, es ist also dx, dy oder dz gleich Null.

■ LREC Punkteliste:x0,y0,z0 ,dx,dy,dz

Dieser Befehl verhält sich wie REC, nur werden zusätzlich die Punkte miteinander verbunden.

■ BOX Punkteliste:x0,y0,z0 ,dx,dy,dz

Mit diesem Befehl werden den 8 Punkten der Liste Koordinaten zugewiesen, so daß ein Quader entsteht. Die linke, untere Ecke liegt im Punkt mit den Koordinaten (x0, y0, z0). Der Quader hat die Maße dx, dy und dz.

■ LBOX Punkteliste:x0,y0,z0 ,dx,dy,dz

Dieser Befehl bewirkt genau dasselbe wie BOX, jedoch wird zusätzlich ein entsprechender LINE-Befehl ausgeführt.

■ MOVE Punkteliste:anfang,ende1[,ende2[,modus]]

Fehlen modus oder ende2, gilt modus=0, ende2=ende1. Innerhalb der Punkteliste kommt es, im Gegensatz zu LINE, nicht auf die Reihenfolge der Punkte an. Wiederholen sich Punkte, hat es auch keinen weiteren Einfluß. Voreinstellung ist »MOVE 0..p-1:0,n-1,n-1,0« (p,n=Anzahl aller Punkte und Bilder). Nach dieser Anweisung werden nur noch die Punkte in der Punkteliste bewegt. Danach folgende Teilanimationen beziehen sich nur noch auf das Bildintervall von »anfang« bis »ende2«, es wird die ganze Bewegung auf die Bilder von »anfang« bis »ende1« verteilt. Danach entscheidet »modus«, wie sie bis »ende2« verlängert wird.

modus=0: Die Bewegung wird genauso weitergeführt.
modus=1: Nach jeweils (ende1 + 1 - anfang) Bildern beginnt die Bewegung von vorne.
modus=2: Nach jeweils (ende1 + 1 - anfang) Bildern kehrt sich die Bewegung um.

Auswirkungen zeigt MOVE auf die Befehle ROT, ROTI, TURN, SHI, SHII, ACC, DEC, EXT.

■ SHI dx,dy,dz

Die Punkte werden linear ansteigend um insgesamt dx, dy und dz verschoben.

■ SHII dx,dy,dz

Hier werden die Punkte ebenfalls um dx, dy und dz verschoben, aber nicht allmählich, sondern in jedem Bild um diesen Betrag.

■ ROT xm,ym,zm,xyw,yzw,dw

Dieser Befehl dient zum Drehen von Objekten und benötigt den Drehpunkt und drei Winkel. Sie müssen also erst einmal die Achse im Raum konstruieren. Der Befehl

```
ROT 200,250,100,0,90,-360
```

bewirkt eine Drehung der Figur um den Punkt (200,250,100). Um die Drehachse zu erhalten, gehen Sie folgendermaßen vor:

Stellen Sie sich eine senkrechte Gerade durch den Drehpunkt vor. Diese wird nun um »xyw« gegen den Uhrzeigersinn parallel zur x-y-Ebene gedreht. Anschließend drehen wir diese Gerade noch um »yzw« Grad (wieder gegen den Uhrzeigersinn) parallel zur y-z-Ebene.

Die Drehung wird auf alle Bilder verteilt. Bei dem obenstehenden Befehl dreht 3DAnim bei insgesamt 120 Bildern das Objekt um jeweils 3 Grad pro Bild.

■ ROTI xm,ym,zm,xyw,yzw,dw

Für die Parameter gilt das gleiche wie beim ROT-Befehl. Der Unterschied ist der, daß in jedem Bild um den Winkel »dw« um die angegebene Achse gedreht wird und nicht wie bei ROT allmählich.

■ TURN xm,ym,zm,xyw,yzw

Auch hier gilt für die Parameter das gleiche wie beim ROT-Befehl, lediglich der Parameter »dw« fehlt. Was also die Bewegung angeht, so wird wie bei ROTI in jedem Bild um bestimmte Winkel gedreht, aber nicht um »dw« um die Achse, die konstruiert wird, sondern so, daß aus der ursprünglichen Achse (parallel zur y-Achse) die Konstruierte entsteht. Das heißt, es werden alle Punkte in jedem Bild um »wx« nach links und um »wz« zum Betrachter gekippt.

■ COPY quelle [,zielanfang] ,zielende

Die Punktkoordinaten des Bildes »quelle« werden auf die der Bilder »zielanfang« bis »zielende« kopiert. Wird der zweite Parameter weggelassen, so ist zielanfang gleich quelle+1 beziehungsweise quelle-1, wenn zielende kleiner als zielanfang ist.

■ COPY quellanfang,quellende,zielanfang,zielende

Die Koordinaten des Intervalls von quellanfang bis quellende werden auf die Bilder von zielanfang bis zielende übertragen.

■ ACC dx,dy,dz

Dieser Befehl entspricht ungefähr der SHI-Anweisung, jedoch wird nicht mit konstanter Geschwindigkeit verschoben, sondern in Ruhelage begonnen und dann konstant beschleunigt.

■ DEC dx,dy,dz

Analog zu ACC wird um dx, dy und dz verschoben, aber mit negativer Beschleunigung, das heißt die Punkte werden immer langsamer, bis sie schließlich ruhen.

■ EXT x0,y0,z0,faktor1,faktor2 [,anfang,ende]

Alle aktuellen Punkte erfahren dadurch eine Streckung mit dem Zentrum (x0,y0,z0). Der Abstand der Punkte dazu wird am Anfang jeweils mit faktor1 multipliziert, ab dann wächst der Faktor linear, bis er am Ende der Bewegung faktor2 ist. Sind die Faktoren -1, so hat man eine Punktspiegelung vor sich. Schaltet man durch den MASK-Befehl eine oder zwei Dimensionen aus, kann man die Streckung auf nur eine Richtung beschränken. Fehlen die letzten beiden Parameter, richtet sich die Bewegung nach dem letzten MOVE-Befehl (von anfang bis ende2).

■ FORM a,b

Die Punkte werden vom Ort im Bild »a« linear zum Ort im Bild »b« überführt. Man kann so zwei ähnliche Figuren definieren und dann ineinander übergehen lassen. Die alten Koordinaten zwischen den Bildern »a« und »b« gehen dabei verloren.

Die Befehle von 3DAnim kennen Sie nun, einige Fragen müssen jedoch noch geklärt werden. 3DAnim benötigt eine Befehlsliste im ASCII-Format, in der die einzelnen Anweisungen stehen, um die vollständige Animation zu berechnen. Die Liste der Befehle muß folgendermaßen aufgebaut sein: In jeder Zeile stehen ein oder mehrere Befehle, die durch einen senkrechten Strich (<SHIFT - \>) zu trennen sind, danach eventuell ein Strichpunkt mit nachfolgendem Kommentar. Jeder dieser Befehle kann mit beliebig vielen REM-Zeichen oder Leerzeichen beginnen. Einem Befehlswort muß ein Trennzeichen folgen (Ziffer, Rechen-, Leerzeichen, runde Klammer), ab da werden dann Leerzeichen und runde Klammern einfach ignoriert. Sie können so Koordinaten optisch zusammenfassen. Ob Sie Buchstaben groß oder klein schreiben, spielt keine Rolle. Wer den Basic-Editor verwenden will, kann durch ein REM-Zeichen (Alt + ä) am Zeilenanfang verhindern, daß der eingegebene Text verändert wird. Die Datei wird am Ende dann mit

```
SAVE "Name",A
```

gespeichert. Nach der Berechnung erzeugt das Programm mehrere Dateien auf Diskette: eine Datei, in der die berechneten Koordi-

naten für die Animation abgelegt sind (Namenserweiterung ».dat«) und Programme in verschiedenen Sprachen, die die Bewegung darstellen. Am Anfang werden Sie gefragt, in welchen Sprachen der Computer diese Programme erzeugen soll. In der abgedruckten Version stehen Ihnen Assembler und Basic zur Verfügung. Geben Sie nun bei der Eingabe eine Zeichenkette ein, die die Zeichen für die gewünschten Sprachen enthält. Sie sind auf dem Bildschirm aufgelistet. Kleinbuchstaben und entsprechende große bewirken dasselbe. Wenn Sie also »a« eingeben, wird Assemblercode erzeugt (Namenserweiterung ».asm«), der danach assembliert werden kann. Bei Eingabe von »b« schreibt Ihr Amiga ein Basic-Programm (Namenserweiterung ».bas«) auf Diskette. Wenn Sie »ab« eingeben, so befinden sich danach beide Programme auf Diskette.

Grundsätzlich versteht 3DAnim als Parameter einen gesamten mathematischen Term (zum Beispiel: -9*23.2+x8/2-v12\10). Zur Verfügung stehen Real-Zahlen (wie 1,4.12,-184,+-.5), Rechenzeichen (+,-,*,/,\), wobei »Punkt vor Strich« gerechnet wird. Außerdem gibt es einige Standardvariablen: »p« und »n« geben die Anzahl der Punkte beziehungsweise der Bilder an. »x«, »y« oder »z« mit nachfolgender Zahl (x0, y4 oder z38) gibt den Wert der entsprechenden Koordinate des Punktes mit dieser Nummer zurück. Dabei wird der Wert aus dem Bild genommen, mit dem die Bewegung beginnt.

Hat ein Parameter den Wert 0, kann man ihn auch weglassen: Aus »SET 1,0,2,0« wird »SET 1,,2«. Man sollte aber bei solchen Befehlen aufpassen, bei denen die Anzahl der Parameter eine Rolle spielt. Ist ein Parameter kleiner 0 oder größer n-1, so wird der Wert mittels »MOD« auf einen Wert dazwischen umgerechnet.

Sie können 3DAnim um andere Sprachen bei der Programmgenerierung erweitern. Wenn Sie etwa C und Pascal hinzufügen wollen, müssen Sie an den DATA-Befehl in Zeile 2 noch »,c,C,p,Pascal« anhängen, des weiteren folgen ab Zeile 3 nun weitere Abfragen, hier also:

```
ELSEIF a$="C" THEN
   CALL C.code
ELSEIF a$="P" THEN
   CALL Pascal.code
```

Beim Vergleich von a$ mit einem Zeichen muß dieses als Großbuchstabe dastehen. Nicht zuletzt ist natürlich der Programmteil nötig, der das Programm in der neuen Sprache auf Diskette schreibt, hier »C.code« und »Pascal.code«. So ein Programmteil wird einfach als Unterprogramm an 3DAnim angefügt. Dabei gibt es einiges zu beachten: Vor dem ersten READ muß der Zeiger durch RESTORE auf den Anfang des Unterprogramms gesetzt werden. Wie das Unterprogramm aufgebaut sein kann, können Sie ja am Beispiel von »Basic.code« sehen und analog programmieren. Wenn Sie keinen Assembler besitzen oder auf Assembler-Code (oder auf Basic-Code) verzichten können, lassen Sie einfach das ganze Unterprogramm weg und ändern noch die Zeilen 2 und 3 um.

Zum Assembler-Code: Ein eventuell erzeugtes Assembler-Programm ist so formuliert, daß es mit dem A68K-Assembler auf der Public Domain-Diskette Fish 110 übersetzt werden kann. Dazu kopieren Sie einfach das Programm (»Name.asm«) in die RAM-Disk und legen die Fish-Disk ein, danach compilieren Sie es vom CLI aus mit dem Befehl

```
a68k/a68k RAM:Name.asm
```

und linken die so entstandene Datei »RAM:Name.o« mit dem Kommando

```
pdc/examples/blink RAM:Name.o to RAM:Name.bin
```

zum lauffähigen Programm »RAM:Name.bin«. Dieses können Sie nun starten, wozu die Datei »Name.dat« auf der eingelegten Diskette stehen muß. Falls Sie diesen Assembler nicht besitzen, dürfte es kein Problem darstellen, 3DAnim entsprechend anzupassen. Falls doch Probleme auftreten, sollten Sie unbedingt die Leerzeichen in den DATA-Zeilen beachten.

Damit das generierte Basic-Programm läuft, müssen im selben Verzeichnis der Amiga-Basic-Interpreter, das Basic-Programm, die Datei mit den Daten und die zwei Dateien »graphics.bmap« und »intuition.bmap« stehen. Wenn Sie einen Amiga mit 512 KByte Speicher besitzen, kann es vorkommen, daß bei den CLEAR-Anweisungen Probleme auftreten. Verkleinern Sie dann die angegebene Größe, bis das Programm läuft. Funktioniert dies nicht, müssen Sie leider die Anzahl der Bilder oder Punkte verringern.

Besitzer eines AC-Basic-Compilers können sich freuen. Sowohl 3DAnim als auch die generierten Programme lassen sich übersetzen. Achten Sie dabei jedoch darauf, daß die Schalter »R« und »N« gesetzt sind.

Zum Testen können Sie Listing 2 verwenden, das ein paar Fähigkeiten von 3DAnim zeigt.

Natürlich soll dieses Programm nicht professionelle ersetzen. Dennoch haben Sie damit eine relativ schnelle und preiswerte Möglichkeit, in die Welt der Animation hineinzuschnuppern. *kn*

Programmname:	3DAnim
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2
Bemerkung:	siehe Text

```
Programmautor: Harry Kellermeyer
--------------------------------

i30 '---------------------          _
GV  '------ 3D-ANIM ------         //
zR  '------  AMIGA  ------      _  //
fA  '- Harry Kellermeyer -      \\ //
Ym  '---------------------       \X/
uo  '----- Init.
LA  CLEAR,150000&
Ku  ON BREAK GOSUB unterbr:BREAK ON
pk  DEFINT i,j
WG  DEF FNst(p1,p2)=(p2<p1)*2+1
HH  DEF FNmd(p1)=(p1 MOD(n+1)+n+1)MOD(n+1)
Dj  DIM ink(5),linie(500),pname$,p,n,xf,yf,flag3,stapel$
nK  DIM v(100),p(8),zeichen0$(15),sprachen0$(15)
V5  pi=4*ATN(1):grad=pi/180
00  stapel$=CHR$(0)
Pq  flag1=-1:flag4=0
jI  fx=1:fy=1:fz=1
Mu  xx=-1:yy=-1:zz=-1
UJ  orgx=0:orgy=0:orgz=0
XB  xf=400:yf=800:zf=1600
op  linie$="":sprachen0=0:zeichen0$="":zeichen1$=""
7P  ink(0)= 0:ink(1)= 0:ink(2)= 0
VH  ink(3)=15:ink(4)=15:ink(5)=15
JH  '----- verfÜgbare Sprachen einlesen
kJ  2 DATA a,Assembler,b,Basic
kk  DATA *,*
eH  READ a$,b$
BY  WHILE a$<>"*"
zc2   sprachen0=sprachen0+1
hW    zeichen0$(sprachen0)=a$
r8    zeichen0$=zeichen0$+UCASE$(a$)
0Q    sprachen0$(sprachen0)=b$
kN    READ a$,b$
zn0 WEND
30  '----- Screen aufbauen u. Programmname u. gewÜnschte Sprach
    en einlesen
y2  SCREEN 1,640,256,2,2:WINDOW 2,,,0,1
qS  PALETTE 0,0,0,1:PALETTE 1,0,0,1
Fr  PALETTE 2,0,0,0:PALETTE 3,1,1,1
iU  COLOR 3,2:CLS
9y  FOR i=0 TO sprachen0-1
1J2   LOCATE 5+(i MOD 5)*2,10+(i\5)*20
X3    PRINT zeichen0$(i+1)"  "sprachen0$(i+1)
mr0 NEXT
En  LOCATE 25,17:INPUT"Name der Quelldatei.................";p
    name$
01  WHILE zeichen1$=""
bz2   LOCATE 27,17:PRINT SPACE$(80)
v1    LOCATE 27,17:INPUT"Sprache(n) der Zieldatei(en).(s.o.).."
      ;zeichen1$
6H    zeichen1$=UCASE$(zeichen1$)
04    IF zeichen1$<>""THEN
Ko4     a$=""
ja      FOR i=1 TO LEN(zeichen1$)
KN6       IF INSTR(zeichen0$,MID$(zeichen1$,i,1))THEN a$=a$+MID
          $(zeichen1$,i,1)
w14     NEXT
iq      zeichen1$=a$
pi2   END IF
L90 WEND
wI  CLS:LOCATE 1,26:PRINT"PROGRAMM: "pname$:PRINT
fa  POKEW &HDFF0*256+&H96,32
Ye  POKEL &HDFF0*256+&H144,0
```

Listing 1. Dreidimensionale Animationen mit »3DAnim«

AMIGA SONDERHEFT

PROGRAMM-SERVICE

Direkt bestellen statt abtippen!

Die aktuellen Disketten zum Sonderheft 9:

Grafik

»Amiga-Objekt-Editor«: Überdimensionale Sprites spielend leicht entwickelt (genaue Programmbeschreibung auf Seite 6 in diesem Heft).
»AOE-Convert«: bauen Sie die mit dem Amiga-Objekt-Editor entwickelten Super-Sprites in Ihre Basic-Programme ein (Beschreibung S.18).
»IFF-Filter«: Farbbilder auf faszinierende Weise modifizieren (Beschreibung S.21).
»Ray Tracer«: Arbeiten mit dem Spiel von Licht und Schatten (Beschreibung S.21).
»3DAnim«: Gittergrafiken – bewegt, perspektivisch, dreidimensional (Beschreibung S.30).

Spiele

Flugsimulator (Beschreibung S.40)
Ping-Pong (Beschreibung S.50)
Krieg der Sterne (Beschreibung S.55)
Denkspiel Trainingslager (Beschreibung S.61)
Strategiespiel »Buh Au CS« (Beschreibung S.67)
Börsenspiel (Beschreibung S.76)

Tools

»MasterCruncher«: Packt alles (Beschreibung S.84).
»Virus Killer«: Unentbehrlich (Beschreibung S.90).
»TestDev«: Welche Devices sind vorhanden? (Beschreibung S.100).
Weiches Ein- und Ausblenden von Bildern (Beschreibung S.101).
»Shuffle«: Der schnelle Weg zum hintersten Fenster (Beschreibung S.102).
»Felder«; Beschleunigtes Abarbeiten von numerischen Feldern (Beschreibung S.104).

Anwendungen

»Keyboard-Master«: Lernprogramm für das 10-Finger-System (Beschreibung S.121).
»Sound-Effekt-Editor«: Geräusche selbst erzeugen (Beschreibung S.124).
Etiketten: Das Programm für Ihre Disketenetiketten (Beschreibung S.130).
»hscroll«: Laufschriften erzeugen (Beschreibung S.136).
»Uni-Datei«: Universelle Dateiverwaltung mit komfortabler Benutzerführung (Beschreibung S.140).

Außerdem finden Sie alle Programme aus den Artikeln, die mit dem D-Symbol gekennzeichnet sind, auf den Programmservice-Disketten.

Bestell-Nr. 45009
Zwei Disketten!

DM 29,90*

sFr 24,90*/öS 299,–*

* Unverbindliche Preisempfehlung



Weitere Angebote auf der Rückseite!

AMIGA PROGRAMMSERVICE

Sie suchen hilfreiche Utilities und professionelle Anwendungen für Ihren Computer? Sie wünschen sich gute Software zu vernünftigen Preisen? Hier finden Sie beides! Unser stetig wachsendes Sortiment enthält interessante Listing-Software für alle gängigen Computertypen. Jeden Monat erweitert sich unser aktuelles Angebot um eine weitere interessante Programmsammlung für jeweils einen Computertyp.
Bei Fragen zu Bestellung und Versand der Programmservice-Disketten wählen Sie bitte:
Telefon (089) 4613-232.

Bestellungen bitte *nur gegen Vorauskasse an:* Markt&Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 4613-0.
Schweiz: Markt&Technik Vertriebs AG, Kollerstrasse 37, CH-6300 Zug, Telefon (042) 440550.
Österreich: Markt&Technik Verlag Gesellschaft m.b.H., Große Neugasse 28, A-1040 Wien, Telefon (0222) 5871393-0; Microcomput-ique, E. Schiller, Göglstraße 17, A-3500 Krems, Telefon (02732) 7493; MES-Versand, Postfach 15, A-3485 Haitzendorf; Bücherzentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Telefon (0222) 833196.
Bestellungen aus anderen Ländern bitte nur schriftlich an: Markt&Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar. Nur gegen Bezahlung der Rechnung im voraus.

Bitte kein Bargeld einschicken!

Bitte verwenden Sie für Ihre Bestellung und Überweisung die beigeheftete Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungs-Scheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

Bit für Bit nur Hits... ...das Beste aus zwei Jahren Amiga-Magazin!

Wir haben auf zwei Disketten die interessantesten Programmangebote der Jahre 1988 und 1989 für die Bereiche Anwendungen/Tools und Spiele gesammelt. Mit dieser Sammlung legen Sie einen leistungsfähigen Grundstein oder bilden eine sinnvolle Erweiterung für Ihre Programmbibliothek. Und dies alles zum besonders günstigen Paketpreis.

DISK I: Anwendungen/Tools

»AmigaDat« – Die Dateiverwaltung für alle Zwecke, von der Schallplattensammlung über Adressen bis hin zur Videosammlung.
»Manager« – Das komfortable Haushaltsbuch.
»Disketi« – Drucken von Diskettenlabels.
»MasterCruncher« – Leistungsfähige Daten- und Programmkomprimierung.
»Recover« – Retten von gelöschten Daten.
»Resi« – Macht Programme resetfest.
»MouseCreator und PointerMaker« – Generieren Sie Ihre eigenen Mauszeiger.
»DiskSpy« – Problemloses Ändern von Daten direkt auf Diskette.
»AmigaSort« – Bringt Ordnung in Ihre Diskettensammlung.
»Fade« – Einfaches Ein- und Ausblenden von Bildschirmen ermöglicht tolle Effekte.
»VirusControl« – Der komfortable Virenschutz.
»TrapHandle« – Die Möglichkeit bei auftretenden Fehlern Ihr C-Programm sauber zu beenden.

DISK II: Die Spielesammlung, die man haben muß

Action
»Troof« – Das spannende Motorradrennen der Zukunft.
»Quadron« – Geschicklichkeit bei höchsten Geschwindigkeiten sind gefragt.

Spieleumsetzungen
»Poker« – Wann bekommen Sie den Royal-Flush.
»Billard« – Tolle Grafik erwartet Sie bei dieser fantastischen Umsetzung.
»Domino« – Verblüffend Umsetzung des bekannten Spiels.
»Kniffel« – Eine grafisch verblüffende Würfelspiel-umsetzung für bis zu vier Spieler.
»3D-Tic-Tac-Toe« – Dreidimensionales Spiel zum Kombinieren und Denken.

Best of the Rest
»Eliza« – Der Amiga als Psychotherapeut.
»Arriba« – Die Tastatur lernt sprechen.

Die Beschreibungen der Programme sind als Readmefile auf den jeweiligen Disketten.
Bestell-Nr. 47901
Zwei Disketten randvoll mit tollen Programmen zum Paketpreis von:
DM 39,90* sFr 39,–*/ öS 390,–*

Direkt bestellen statt abtippen! Die aktuelle Diskette zum Heft: Amiga-Sonderheft 7

Reflex: Das neueste Projekt von Arno Gölzer. »Reflex« ist der Nachfolger der beliebten Billardumsetzung. Lassen Sie sich von einer einzigartigen Spielidee und der tollen grafischen Umsetzung begeistern. *Beschreibung auf Seite 6ff.* Noch **mehr starke Spiele...** Ob Action- oder Strategiespiel — hier finden Sie alles, was das Spielerherz begehrt. **»Fast Freddie«** *Die Beschreibung dieses Actionspiels finden Sie auf Seite 86.* **»Memory«** Eine interessante grafische Umsetzung für Leute mit gutem Gedächtnis. *Beschreibung auf Seite 97.* **»Reactor«** Das Strategiespiel, das auch Atomkraftgegner fesseln wird. *Beschreibung auf Seite 107.* **»Schiebespiel«** Strategisches Denken und Tüfteln ist bei diesem Spiel das A und O. *Beschreibung auf Seite 88.*
Vector-Grafik-Library: Nutzen Sie die ganze Faszination der Vektor-Grafik. Mit dieser Bibliothek bauen Sie Vector-Grafiken auf einfache und schnelle Weise in Ihre Programme ein. Der Vorteil: Es wird nur wenig Speicherplatz und Rechenzeit verbraucht. *Beschreibung auf Seite 22ff.*
Zwei Disketten
Bestell-Nr. 45907 **DM 34,90*** sFr 29,50*/öS 349,–*

Amiga-Sonderheft 2 Grafik, Anwendung

Object-Editor: Animierte Figuren, beispielsweise für eigene Spiele, entwickeln Sie mit diesem Editor auf komfortable Weise. Sogar mit Deluxe Paint erstellte Pinsel lassen sich einlesen.
Haushaltsbuch: Mit diesem hervorragenden Anwendungsprogramm verwalten Sie alle Einnahmen und Ausgaben auf übersichtliche Weise. Eine Monats- oder Jahresstatistik zeigt, in welchen Bereichen Sie zukünftig sparen können. Jetzt haben Sie Ihre Finanzen im Griff.
Keyboard-Master: Lernen Sie im Zehn-Finger-System zu tippen. Mit diesem didaktisch ausgereiften Programm ist dies kein Problem. Für Programmierer sind sogar Spezial-Lektionen mit wichtigen Sonderzeichen vorhanden.
FastLoadCopy: Dieses Tool bringt den DIR-Befehl auf Trab. Nach der »Operation« wird das Inhaltsverzeichnis einer Diskette im D-Zug-Tempo eingelesen. Zusätzlich kopiert das Programm Disketten und versieht diese mit dem schnellen Directory.
Weiterhin befinden sich auf der Diskette alle Programme, die im Inhaltsverzeichnis des Amiga-Sonderhefts 2 mit einem Diskettensymbol gekennzeichnet sind.
3,5"-Diskette für Amiga
Bestell-Nr. 45802 **DM 29,90*** sFr 24,90*/öS 299,–*

Die Wiederbelebung für die C64-Peripherie

Viele Amiga-Besitzer haben noch einen C64 mit Peripheriegeräten zu Hause stehen. Mit ein bißchen Hard- und Software können Sie diese zu neuem Leben erwecken und Ihre Daten so weiterbenutzen. Dabei ist die Bedienung wirklich einfach.
Der fertig aufgebaute IEC-Handler erlaubt es, alle C64-Geräte wie die Floppy 1541 oder 1571, Commodore-MPS-Drucker und natürlich auch den C64 (zur Datenübertragung) am Amiga zu betreiben.
Das Gesamtpaket besteht aus der fertig aufgebauten Platine mit Verbindungskabel, der Treibersoftware auf 3,5"-Diskette sowie einer entsprechenden Dokumentation.
Bestell-Nr. 39101 **DM 79,–*** sFr 71,–*/öS 790,–*

Amiga-Sonderheft 8 Spiel Nr. 1: »Maze«

»Maze« ist ist das Spiel für Denker und Lenker. Schnelles Auffassungsvermögen und eine ausgeklügelte Taktik entscheiden dieses Spiel. Ein langes Spielvergnügen ist durch den eingebauten Level-Editor garantiert.
Weitere Informationen über dieses faszinierende Spiel finden Sie auf Seite 106 in diesem Heft.

Spiel Nr. 2: »Oel-Fieber«

Steigen Sie ein in das Millionengeschäft um Öl und Aktien. Ihre Fähigkeiten als Manager werden auf immer neue Bewährungsproben gestellt. Das amüsante Würfelspiel erfordert dabei eine Menge Kreativität, um immer neue Strategien zu entwickeln. Wie Sie mit Öl- und Aktiengeschäften reich werden können, lesen Sie in diesem Heft ab Seite 89.

Sicher ist sicher

Unumgänglich für jeden Amiga-User ist ein leistungsfähiges Anti-Viren-Programm. »Virus-Control v 1.4« schützt Sie vor allen bekannten Bootblock- und Linkviren sowie durch seine Lernfähigkeit auch vor zukünftigen »Infektionskrankheiten«. Schaffen Sie sich mit »Virus-Control v 1.4« ein zuverlässiges Immunsystem für Ihren Amiga. Die Anleitung finden Sie auf der Diskette.
Bestell-Nr. 45908 **DM 29,90*** sFr 24,90*/öS 299,–*

* Unverbindliche Preisempfehlung

Wichtig: Mit den Gutscheinen aus dem »Super-Software-Scheckheft« zu DM 149,– können Sie Software-Disketten Ihrer Wahl aus dem Programmservice-Angebot im Wert von DM 180,– bestellen – egal, ob diese DM 19,90, DM 29,90 oder DM 89,– kosten. Sie sparen DM 30,–!

Das Super-Software-Angebot finden Sie in den Zeitschriften

Computer Persönlich, PC Magazin Plus, Amiga-Magazin, Amiga-Sonderheft, 64'er-Magazin, 64'er-Sonderheft, ST-Magazin, PC Magazin, Computer live.

Übrigens: Die Gutscheine können Sie auch übertragen oder verschenken!

Das Scheckheft können Sie per Verrechnungsscheck oder mit der eingehefteten Zahlkarte direkt beim Verlag bestellen.
Kennwort: »Super-Software-Scheckheft«, Bestell-Nr. W156

```
'----- Interpreterschleife
OPEN pname$ FOR INPUT AS 1
WHILE NOT EOF(1)
  LINE INPUT#1,a$:a$=UCASE$(LEFT$(a$,INSTR(a$+";",";")-1))
  :zeile$=a$
  WHILE zeile$<>""
    a$=LEFT$(zeile$,INSTR(zeile$+"|","|")-1):zeile$=MID$(ze
    ile$,LEN(a$)+2)
    b$=LEFT$(a$,1):WHILE b$=" "OR b$="'":a$=MID$(a$,2):b$=L
    EFT$(a$,1):WEND
    befehl$=a$:IF a$<>""THEN GOSUB ausfuehren
  WEND
WEND
CLOSE
'----- Daten der Punkte umrechnen + speichern und Programme
 erzeugen
GOSUB umrechnen:GOSUB linie.interpr
OPEN pname$+".dat" FOR OUTPUT AS 1
FOR j=0 TO n
  FOR i=0 TO p:PRINT#1,MKI$(x(i,j));MKI$(y(i,j));:NEXT
NEXT
CLOSE
FOR i=1 TO LEN(zeichen0$)
  a$=MID$(zeichen0$,i,1)
  a=INSTR(zeichen1$,a$)
  IF a THEN
    IF a$="A" THEN
      GOSUB Assembler.code
    ELSEIF a$="B" THEN
      GOSUB Basic.code
    'Hier können weitere Abfragen folgen.
    ELSE
    END IF
  END IF
NEXT i
'----- Starten des erzeugten Basic-Programms oder Ende
GOSUB schliessen
REM IF INSTR(zeichen1$,"B")THEN RUN pname$+".bas"
END
'***** Abbruch des Programms
'----- Unterbrechung
unterbr:
  GOSUB schliessen
END
'----- Fehlermeldung
f:
  PRINT:PRINT"Fehler: ";:BEEP
  COLOR 2,3:PRINT befehl$:COLOR 3,2
  WHILE INKEY$="":WEND
  GOSUB schliessen
END
'***** Unterprogramme (GOSUB) :
'----- Fenster + Screen schließen
schliessen:
  WINDOW CLOSE 2
  SCREEN CLOSE 1
RETURN
'----- Befehl ausfuehren
ausfuehren:
  GOSUB untersuchen
  PRINT "  " b$ TAB(10) a$
  b$=LEFT$(b$+".....",5)
  c$=  "DIM..PERS.SET..ROTI.ROT..LINE.MOVE.SHI..SHII.TURN.
  COPY.RES.."
  c$=c$+"ORIG.MASK.ACC..DEC..FORM.EXT..LET..BOX..LBOX.REC..
  LREC.INKS."
  a=(INSTR(c$,b$)+4)/5
  IF a<1 OR a>1 AND flag1 THEN f
  IF RIGHT$(a$,1)=","THEN a$=a$+"0"
  c$=a$
  ON a GOSUB c.dim,c.pers,c.set,c.roti,c.rot,c.line,c.move,
  c.shi,c.shii,c.turn,c.copy,c.res,c.orig,c.mask,c.acc,c.acc
  ,c.form,c.ext,c.let,c.box,c.lbox,c.rec,c.lrec,c.inks
RETURN
'----- a$ auf zulässige Zeichen prÜfen
untersuchen:
  b$="":c$=LEFT$(a$,1)
  WHILE "A"<=c$ AND c$<="Z"
    b$=b$+c$:a$=MID$(a$,2):c$=LEFT$(a$,1)
  WEND
  IF b$=""OR LEN(b$)>4 THEN f
  c$=""
```

Listing 1. Dreidimensionale Animationen mit »3DAnim« (Fortsetzung)

Impressum

Herausgeber: *Carl-Franz von Quadt, Otmar Weber*
Redaktionsdirektor: *Richard Kerler*
Chefredakteur: *Wolfram Höfler*
Stellv. Chefredakteur: *Gottfried Knechtel (kn) – verantwortlich für den redaktionellen Teil*
Chef vom Dienst: *Susanne Kirmaier*
Mitarbeiter dieser Ausgabe: *René Beaupoil, Uta Jürgens*
Redaktionsassistenz: *Brigitte Bobenstetter, Sylvia Derenthal, Helga Weber (202)*
Telefax: *089/4613-778, Hotline: Donnerstag 14 bis 17 Uhr*
Alle Artikel sind mit dem Kurzzeichen des Redakteurs und/oder mit dem Namen des Autors/Mitarbeiters gekennzeichnet

Manuskripteinsendungen: *Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt&Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt&Technik Verlag AG verlegten Publikationen. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.*

Art-director: *Friedemann Porscha*
Titelgestaltung: *Rolf Boyke*
Layout: *Alexander Kowarzyk (Cheflayouter), Andrea Miller*
Bildredaktion: *Janos Feitser (Ltg.), Sabine Tennstaedt; Roland Müller (Fotografie); Ewald Standke, Norbert Raab (Spritzgrafik); Werner Nienstedt (Computergrafik)*
Bildnachweis: *Südd. Zeitung (Titelmotiv)*

Anzeigenleitung: *Alicia Clees (313) – verantwortlich für die Anzeigen*
Telefax: *089/4613-775*

Anzeigenverwaltung und Disposition: *Monika Burseg (147)*

Auslandsrepräsentation:
Anzeigenverkaufsleitung Ausland: *Thomas Schlüter (299)*
Auslandsniederlassungen:
Schweiz: *Markt&Technik Vertriebs AG, Kollerstr. 37, CH-6300 Zug, Tel. 042-440550/660, Fax 042-415770, Telex: 862329 mut ch*
USA: *M&T Publishing Inc.; 501 Galveston Drive Redwood City, CA 94063, Telefon: (415) 366-3600, Telex 752-351*
Österreich: *Markt&Technik Ges. mbH, Große Neugasse 28, A 1040-Wien, Telefon: 0222/5871393, Telex: 047-132532*
Anzeigen-Auslandsvertretung:
England: *F. A. Smyth&Associates Limited, 23a, Aylmer Parade, London, N2 OPQ. Telefon:0044/1/3405058, Telefax: 0044/1/3419602*
Israel: *Baruch Schaefer, Haeskel-Str. 12, 58348 Holon, Israel, Tel. 00972-3-5562256*
Taiwan: *Aim International Inc., 4F.-1, No. 200, Sec. 2, Hsin-I Rd.; Taipei, Taiwan, R.O.C., Tel. 00886-2-7548631, -7548633, Fax 00886-2-7548710*
Korea: *Young Media Inc., C.P.O. Box: 6113, Seoul/Korea, Tel. 0082-2-7564819, /-7742759, Fax 0082-2-7575789*

Vertriebsleitung: *Helmut Grünfeldt (189)*

Vertrieb Handelsauflage: *Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: ip Internationale Presse, Hauptstätter Straße 96, 7000 Stuttgart 1, Tel. 0711/6483-110*

Bezugsmöglichkeiten: *Leser-Service: Telefon (089) 4613-366. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.*

Preis: *Das Einzelheft kostet DM 16,–*

Produktion: *Technik: Klaus Buck (Ltg./180), Wolfgang Meyer (Stellv./887); Herstellung: Otto Albrecht (Ltg./917)*

Druck: *SOV Graphische Betriebe, Laubanger 23, 8600 Bamberg*

Sonderdruck-Dienst: *Alle in dieser Ausgabe erschienenen Beiträge sind in Form von Sonderdrucken zu erhalten. Anfragen an Reinhard Jarczok, Tel. 089/4613-185, Fax 4613-776.*

Vorstand: *Otmar Weber (Vors.), Bernd Balzer, Richard Kerler*

Verlagsdirektor: *Wolfram Höfler*

Direktor Zeitschriften: *Michael M. Pauly*

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen: *Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 522052, Telefax 089/4613-100*

Telefon-Durchwahl im Verlag: *Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089/4613 und dann die Nummer, die in den Klammern hinter dem jeweiligen Namen angegeben ist.*

```
  IF a$<>""THEN
    c$=""
    FOR j=1 TO LEN(a$):e$=MID$(a$,j,1)
      IF INSTR("XYZPNV:,.+-*/\0123456789 ()",e$)=0 THEN f E
      LSE IF INSTR("( )",e$)=0 THEN c$=c$+e$
    NEXT
    a$=c$
  END IF
RETURN
'----- neues k berechnen
k:
  k=k+r:a=(k=m+1 OR k=-1)
  IF mo=2 AND a THEN k=k-r-r:r=-r
  IF mo=1 AND a THEN k=1
RETURN
'----- x0,y0,z0 holen
x0y0z0:
  flag4=-1:GOSUB h:x0=a:GOSUB h:y0=a:GOSUB h:z0=a
RETURN
'----- addieren bei DEC,ACC,SHI,SHII
add:
  IF xx THEN x(i,j)=x(i,j)+x0*a
  IF yy THEN y(i,j)=y(i,j)+y0*a
  IF zz THEN z(i,j)=z(i,j)+z0*a
RETURN
'----- Parameter von c$ nach a holen
h:
  s=0:s2=0:f$=""
  d=INSTR(c$,","):IF d THEN f$=","
  a=INSTR(c$,":"):IF a AND(a<d OR d=0)THEN f$=":":d=a
  a=INSTR(c$,".."):IF a AND(a<d OR d=0)THEN f$="..":d=a
  IF d THEN d$="+"+LEFT$(c$,d-1):c$=MID$(c$,d+LEN(f$))ELSE
  d$="+"+c$:c$=""
  IF d$<>"+"THEN
  WHILE d$<>""
    f$=LEFT$(d$,1):GOSUB objekt.holen
    IF f$="+"THEN
      s=s+s2:s2=a
    ELSEIF f$="-"THEN
      s=s+s2:s2=-a
    ELSEIF f$="*"THEN
      s2=s2*a
    ELSEIF f$="/"THEN
      IF a=0 THEN f ELSE s2=s2/a
    ELSEIF f$="\"THEN
      IF a=0 THEN f ELSE s2=s2\a
    ELSE
      GOTO f
    END IF
  WEND
  END IF
  a=s+s2
RETURN
objekt.holen:
  d$=MID$(d$,2):e$=LEFT$(d$,1):IF e$=""THEN f
  v=1
  WHILE e$="+"OR e$="-"
    IF e$="-"THEN v=-v
    d$=MID$(d$,2):e$=LEFT$(d$,1)
  WEND
  IF INSTR(".0123456789",e$)THEN  'Zahl
    GOSUB ziffern.holen:a=d
  ELSEIF e$="N"THEN               'Anzahl der Bilder
    a=n+1:d$=MID$(d$,2)
  ELSEIF e$="P"THEN               'Anzahl der Punkte
    a=p+1:d$=MID$(d$,2)
  ELSEIF INSTR("XYZ",e$)THEN      'Koordinate
    g$=e$:d$=MID$(d$,2):GOSUB ziffern.holen
    IF g$="X"THEN a=x(d,an)ELSE IF g$="Y"THEN a=y(d,an)ELSE
     a=z(d,an)
  ELSEIF e$="V"THEN               'Variable
    d$=MID$(d$,2):GOSUB ziffern.holen:a=v(d)
  ELSE
    GOTO f
  END IF
  a=v*a
RETURN
zahl.holen:
  a=1
  WHILE e$="+"OR e$="-"
    IF e$="-"THEN a=-a
    d$=MID$(d$,2):e$=LEFT$(d$,1)
  WEND
  GOSUB ziffern.holen:a=a*d
RETURN
ziffern.holen:
  komma=0:e$="":h$=LEFT$(d$,1):IF h$=""OR INSTR(".012345678
  9",h$)=0 THEN f
  WHILE INSTR(".0123456789",h$)AND d$<>""
    IF h$="."THEN IF komma THEN f ELSE komma=-1
    e$=e$+h$:d$=MID$(d$,2):h$=LEFT$(d$,1)
  WEND
  d=VAL(e$)
RETURN
'----- (i,j)-ten Punkt um wx und wz drehen
drehen:
  IF wz THEN
    d1=z(i,j)-z0:d2=y(i,j)-y0:GOSUB winkel:w=w+wz*v
    z(i,j)=z0+COS(w)*b
    y(i,j)=y0+SIN(w)*b
  END IF
  IF wx THEN
    d1=x(i,j)-x0:d2=y(i,j)-y0:GOSUB winkel:w=w+wx*v
    x(i,j)=x0+COS(w)*b
    y(i,j)=y0+SIN(w)*b
  END IF
RETURN
'----- Parameter fÜr Rotationsbefehle holen
get.rot:
  GOSUB x0y0z0:GOSUB h:wx=a*grad:GOSUB h:wz=a*grad
  flag6=c$<>"":GOSUB h:dw=a*grad:flag7=c$<>""
  IF flag7 THEN
    x1=wx/grad:y1=wz/grad:z1=dw/grad
    GOSUB h:dw=a*grad
  END IF
RETURN
'----- 3D- in 2D-Koordinaten umrechnen
umrechnen:
  xf=(xf+orgx)/fx:yf=(yf+orgy)/fy:zf=(zf+orgz)/fz
  FOR j=0 TO n
    FOR i=0 TO p
      a=(z(i,j)+orgz)/fz
      x(i,j)=    (a*xf+(zf-a)*(x(i,j)+orgx)/fx)/zf
      y(i,j)=255-(a*yf+(zf-a)*(y(i,j)+orgy)/fy)/zf/2
    NEXT
  NEXT
RETURN
'----- Eingabe: d1,d2 = Koord. e. Vektors
'      Ausgabe: w = Zwischenwinkel, b = Betrag
winkel:
  IF d1=0 THEN
    w=pi/2:IF d2<0 THEN w=w*3
  ELSE
    w=ATN(d2/ABS(d1)):IF d1<0 THEN w=pi-w
  END IF
  b=SQR(d1*d1+d2*d2)
RETURN
'----- Eingabe: x0,y0,z0, x1,y1,z1
'      Ausgabe: wx,wz
trans:
  IF x1-x0 OR y1-y0 OR z1-z0 THEN ELSE f
  d1=z1-z0:d2=y1-y0:GOSUB winkel:wz=w-pi/2
  d1=x1-x0:d2=    b:GOSUB winkel:wx=w-pi/2
RETURN
'----- linie$ interpretieren
linie.interpr:
  a$=MID$(linie$,2):linie(1)=1:j=2 'Move
  WHILE a$<>""
    a=INSTR(a$+":",":"):c$=LEFT$(a$,a-1):a$=MID$(a$,a+1)
    IF RIGHT$(c$,1)=","THEN c$=c$+"0"
    WHILE c$<>""
      GOSUB intervall.holen
      IF d=0 THEN m2=m1
      FOR i=m1 TO m2 STEP FNst(m1,m2)
        linie(j)=i:linie(j+1)=2:j=j+2 'Draw
      NEXT
    WEND
    linie(j-1)=1 'doch Move
  WEND
  linie(0)=j-3
RETURN
intervall.holen:
  a=INSTR(c$,".."):b=INSTR(c$,",")
  IF a=0 OR a>b AND b THEN
    GOSUB h:m1=a:d=0
```

```
  ELSE
    IF a=1 THEN m1=0:c$=MID$(c$,3)ELSE GOSUB h:m1=a
    IF LEFT$(c$,1)=","OR c$=""THEN m2=p:c$=MID$(c$,2)ELSE GOSUB h:m2=a
    d=-1
  END IF
RETURN
'***** Unterprogramme für die einzelnen Befehle:
c.dim:
  flag1=NOT flag1:IF flag1 THEN f
  GOSUB h:p=a-1:GOSUB h:n=a-1:flag3=c$=""
  DIM x(p,n),y(p,n),z(p,n),punkt(p)
  FOR i=0 TO p:punkt(i)=-1:NEXT
  an=0:en=n:en2=n:mo=0
RETURN
c.pers:
  GOSUB h:xf=a:GOSUB h:yf=a:GOSUB h:zf=a
RETURN
c.set:
  GOSUB h:set=a:a=INSTR(c$+":",":")
  b$=MID$(c$,a+1):c$=LEFT$(c$,a-1)
  WHILE c$<>""
    GOSUB x0y0z0:a=0:m2=n
    IF c$<>""THEN GOSUB h:m2=a:GOSUB h
    FOR i=a TO m2 STEP FNst(a,m2)
      IF xx THEN x(set,i)=x0
      IF yy THEN y(set,i)=y0
      IF zz THEN z(set,i)=z0
    NEXT
    set=set+1:a=INSTR(b$+":",":")
    c$=LEFT$(b$,a-1):b$=MID$(b$,a+1)
  WEND
RETURN
c.move:
  ERASE punkt:DIM punkt(p)
  a=INSTR(c$,":"):IF a<2 THEN f
  b$=MID$(c$,a+1):c$=LEFT$(c$,a-1)
  IF RIGHT$(c$,1)=","THEN c$=c$+"0"
  WHILE c$<>""
    GOSUB intervall.holen
    IF d=0 THEN m2=m1
    FOR i=m1 TO m2 STEP FNst(m1,m2):punkt(i)=-1:NEXT
  WEND
  c$=b$:GOSUB h:an=a
  en =an:IF c$<>""THEN GOSUB h:en=a
  en2=en:IF c$<>""THEN GOSUB h:en2=a
  GOSUB h:mo=a
  IF en<an OR en2<an OR ABS(mo-1)>1 THEN f
  IF en2=en THEN mo=0
RETURN
c.line:
  linie$=linie$+":"+a$
RETURN
c.rot:
  GOSUB get.rot
  IF flag7 THEN GOSUB trans
  m=en-an+1
  FOR i=0 TO p
    IF punkt(i)THEN
      k=1:r=1
      FOR e=an TO en2:j=FNmd(e)
        v=-1:GOSUB drehen
        d1=x(i,j)-x0:d2=z(i,j)-z0
        GOSUB winkel:w=w+dw/m*k
        x(i,j)=x0+COS(w)*b
        z(i,j)=z0+SIN(w)*b
        v=1:GOSUB drehen:GOSUB k
      NEXT
    END IF
  NEXT
RETURN
c.roti:
  Push en:Push mo
    en=an:mo=1:GOSUB c.rot
  Pop mo:Pop en
RETURN
c.turn:
  GOSUB get.rot
  IF flag6 THEN x1=wx/grad:y1=wz/grad:z1=dw/grad:GOSUB trans
  FOR i=0 TO p
   IF punkt(i)THEN
     FOR e=an TO en2
       j=FNmd(e):v=1:GOSUB drehen
     NEXT
   END IF
  NEXT
RETURN
c.shi:
  GOSUB x0y0z0:m=en-an+1
  FOR i=0 TO p
    IF punkt(i)THEN
      k=1:r=1
      FOR e=an TO en2
        j=FNmd(e):a=k/m:GOSUB add:GOSUB k
      NEXT
    END IF
  NEXT
RETURN
c.shii:
  Push en:Push mo
    en=an:mo=1:GOSUB c.shi
  Pop mo:Pop en
RETURN
c.copy:
  flag4=-1
  GOSUB h:m1=a:GOSUB h:m2=a
  IF c$<>""THEN GOSUB h:m3=a ELSE m3=m2:m2=m1+FNst(m1,m2)
  IF c$<>""THEN
    GOSUB h:b=FNst(m1,m2):c=FNst(m3,a)
    IF ABS((m2-m1)/(a-m3))<>1 THEN f
    FOR i=0 TO p
      IF punkt(i)THEN
        FOR e=m1 TO m2 STEP b
          j=FNmd(e)
          k=FNmd((e-m1)/b*c+m3)
          IF xx THEN x(i,k)=x(i,j)
          IF yy THEN y(i,k)=y(i,j)
          IF zz THEN z(i,k)=z(i,j)
        NEXT
      END IF
    NEXT
  ELSE
    FOR i=0 TO p
      IF punkt(i)THEN
        FOR e=m2 TO m3 STEP FNst(m2,m3)
          j=FNmd(e)
          IF xx THEN x(i,j)=x(i,m1)
          IF yy THEN y(i,j)=y(i,m1)
          IF zz THEN z(i,j)=z(i,m1)
        NEXT
      END IF
    NEXT
  END IF
RETURN
c.res:
  x0=fx:y0=fy:z0=fz
  fx= 1:IF c$<>""THEN GOSUB h:fx=a/640
  fy=fx:IF c$<>""THEN GOSUB h:fy=a/512
  fz=fx:IF c$<>""THEN GOSUB h:fz=a/640
  IF fx*fy*fz=0 THEN f
  xf=xf/x0*fx:yf=yf/y0*fy:zf=zf/z0*fz
  orgx=orgx/x0*fx:orgy=orgy/y0*fy:orgz=orgz/z0*fz
  IF flag4 THEN
    FOR i=0 TO p
      FOR j=0 TO n
        x(i,j)=x(i,j)/x0*fx
        y(i,j)=y(i,j)/y0*fy
        z(i,j)=z(i,j)/z0*fz
      NEXT
    NEXT
  END IF
  flag4=-1
RETURN
c.orig:
  x0=orgx:y0=orgy:z0=orgz
  GOSUB h:orgx=a:GOSUB h:orgy=a:GOSUB h:orgz=a
  IF flag4 THEN
    FOR i=0 TO p
      FOR j=0 TO n
        x(i,j)=x(i,j)+x0-orgx
        y(i,j)=y(i,j)+y0-orgy
        z(i,j)=z(i,j)+z0-orgz
      NEXT
    NEXT
  END IF
  xf=xf+x0-orgx:yf=yf+y0-orgy:zf=zf+z0-orgz
RETURN
```

Listing 1. Dreidimensionale Animationen mit »3DAnim« (Fortsetzung)

```
C4  c.mask:
ON2   GOSUB h:xx=a:GOSUB h:yy=a:GOSUB h:zz=a
pR0 RETURN
Fm  c.acc:
UH2   flag2=LEFT$(b$,3)="DEC"
6b    GOSUB x0y0z0:m=en-an+1
ew    FOR i=0 TO p
U44     IF punkt(i)THEN
Q16       k=1:IF flag2 THEN k=m-1
8F        FOR e=an TO en2
1I8         j=FNmd(e)
7Q          a=k*k/m/m:k=k+1
fL          IF flag2 THEN a=1-a:k=k-2
3A          GOSUB add
kp6       NEXT
cV4     END IF
mr2   NEXT
4g0 RETURN
8D  c.form:
J12   GOSUB h:m1=a:GOSUB h:m2=a:m=ABS(m2-m1)
sA    FOR i=0 TO p
1I4     IF punkt(i)THEN
Vh6       x0=x(i,m2)-x(i,m1)
gv        y0=y(i,m2)-y(i,m1)
r9        z0=z(i,m2)-z(i,m1)
Bv        a=FNst(m1,m2):k=1
vF        FOR e=m1+a TO m2-a STEP a:j=FNmd(e)
d08         IF xx THEN x(i,j)=x(i,m1)+x0*k/m
01          IF yy THEN y(i,j)=y(i,m1)+y0*k/m
N8          IF zz THEN z(i,j)=z(i,m1)+z0*k/m
75          k=k+1
166       NEXT
tm4     END IF
382   NEXT
Lx0 RETURN
MC  c.ext:
OR2   a$=c$:GOSUB x0y0z0:GOSUB h:GOSUB h:Push an:Push en2
JQ    IF c$<>""THEN GOSUB h:an=a:GOSUB h:en2=m2
uJ    c$=a$
eI    GOSUB x0y0z0:GOSUB h:m1=a:GOSUB h:m2=a
QR    m3=an:m4=en2:IF c$<>""THEN GOSUB h:m3=a:GOSUB h:m4=a:IF
       m4<m3 THEN f
DV    FOR i=0 TO p
3d4     IF punkt(i)THEN
En6       FOR e=m3 TO m4:j=FNmd(e)
6k8         a=m1+(m2-m1)/(m4-m3)*(e-m3)
io          IF xx THEN x(i,j)=x0+(x(i,j)-x0)*a
CE          IF yy THEN y(i,j)=y0+(y(i,j)-y0)*a
ge          IF zz THEN z(i,j)=z0+(z(i,j)-z0)*a
IN6       NEXT
A34     END IF
KP2   NEXT
7F    Pop en2:Pop an
dF0 RETURN
ZD  c.let:
Xr2   IF LEFT$(c$,1)<>"V"THEN f
Ry    a=INSTR(c$+",",","):d$=MID$(c$,2,a-2):c$=MID$(c$,a+1)
yD    GOSUB ziffern.holen:m1=d:GOSUB h:v(m1)=a
1K0 RETURN
Ca  c.box:
CY2   GOSUB daten.holen:IF p(0)-8 THEN f
Di    FOR e=m1 TO m2:j=FNmd(e)
1w4     a=1:x1=x0:y1=y0:z1=z0:GOSUB s
d6      x1=x0+dx:GOSUB s
0Z      z1=z0+dz:GOSUB s
6k      x1=x0:GOSUB s
wp      y1=y0+dy:z1=z0:GOSUB s
hA      x1=x0+dx:GOSUB s
4d      z1=z0+dz:GOSUB s
Ao      x1=x0:GOSUB s
di2   NEXT
vX0 RETURN
WH  s:
Lx2   IF xx THEN x(p(a),j)=x1
dE    IF yy THEN y(p(a),j)=y1
vV    IF zz THEN z(p(a),j)=z1
sW    a=a+1
1d0 RETURN
Ar  daten.holen:
QB2   ERASE p:DIM p(8)
Zr    a=INSTR(c$,":"):IF a<2 THEN f
hq    b$=MID$(c$,a+1):c$=LEFT$(c$,a-1)
f7    IF RIGHT$(c$,1)=","THEN c$=c$+"0"
4J    WHILE c$<>""
104     GOSUB intervall.holen
wi      IF d=0 THEN m2=m1
rD      FOR i=m1 TO m2 STEP FNst(m1,m2):p(0)=p(0)+1:p(p(0))=i:N
        EXT
G42   WEND
mC    c$=b$
L2    GOSUB x0y0z0:GOSUB h:dx=a:dy=a:dz=a:m1=0:m2=n
W0    IF c$<>""THEN GOSUB h:dy=a
c7    IF c$<>""THEN GOSUB h:dz=a
Kn    IF c$<>""THEN GOSUB h:m1=a
Ks    IF c$<>""THEN GOSUB h:m2=a:IF m2<m1 THEN f
Iu0 RETURN
16  c.lbox:
Js2   GOSUB c.box
GX    c$="1234156785:26:37:48":GOSUB anhaengen
My0 RETURN
fp  anhaengen:
Sb2   linie$=linie$+":"
H7    FOR i=1 TO LEN(c$)
e54     a=VAL(MID$(c$,i,1))
CL      IF a THEN
Ni6       linie$=linie$+MID$(STR$(p(a)),2)+","
1R4     ELSE
HD6       MID$(linie$,LEN(linie$),1)=":"
5y4     END IF
FK2   NEXT
4k    linie$=LEFT$(linie$,LEN(linie$)-1)
YA0 RETURN
E4  c.rec:
f92   GOSUB daten.holen:IF p(0)-4 OR dx*dy*dz THEN f
3Y    FOR e=m1 TO m2:j=FNmd(e)
bm4     a=1:x1=x0:y1=y0:z1=z0:GOSUB s
5U      x1=x0+dx:IF dx=0 THEN z1=z0+dz
hu      GOSUB s
Kg      y1=y0+dy:IF dx THEN z1=z0+dz
jw      GOSUB s
Ul      x1=x0:IF dx=0 THEN z1=z0
ly      GOSUB s
SX2   NEXT
kM0 RETURN
rl  c.lrec:
PX2   GOSUB c.rec
1A    c$="12341":GOSUB anhaengen
oQ0 RETURN
VA  c.inks:
Q12   FOR i=1 TO 0 STEP-1
aF4     GOSUB x0y0z0
1Z      ink(0+i*3)=x0:ink(1+i*3)=y0:ink(2+i*3)=z0
ch2   NEXT
uW0 RETURN
R9  '----- Basicprogram auf Diskette erzeugen
Lm  Basic.code:
nY  Basic.code.data:RESTORE Basic.code.data
at2   OPEN pname$+".bas" FOR OUTPUT AS 1
gq    PRINT#1,"LIBRARY"+CHR$(34)+"intuition.library"
Wa    PRINT#1,"LIBRARY"+CHR$(34)+"graphics.library"
bh    PRINT#1,"CLEAR,100000"
fN    Read.data
Ut    DATA "SCREEN 2,640,256,1,2","WINDOW 2,,,0,2"
Oy    DATA scr2&=PEEKL(WINDOW(7)+46),bm2&=PEEKL(WINDOW(8)+4),*
54    PRINT#1,"PALETTE 0,"+STR$(ink(0)/15)+","+STR$(ink(1)/15)
      +","+STR$(ink(2)/15)
Sb    PRINT#1,"PALETTE 1,"+STR$(ink(3)/15)+","+STR$(ink(4)/15)
      +","+STR$(ink(5)/15)
kS    Read.data
Qn    DATA "SCREEN 1,640,256,1,2","WINDOW 2,,,0,1"
TX    DATA rp&=WINDOW(8),bm1&=PEEKL(rp&+4),*
wT    PRINT#1,"OPEN"+CHR$(34)+pname$+".dat"+CHR$(34)+"FOR INPU
      T AS 1"
OE    PRINT#1,"p%="+STR$(p)+":n%="+STR$(n)+":DIM x%(p%,n%),y%(
      p%,n%)"
Ju    PRINT#1,"xf%="+STR$(xf)+":yf%="+STR$(255-yf/2)"
Tj    PRINT#1,"FOR j%=0TO n%:FOR i%=0TO p%:x%(i%,j%)=CVI(INPUT
      $(2,1)):y%(i%,j%)=CVI(INPUT$(2,1)):NEXT:NEXT"
1D    PRINT#1,"j%=0:ScreenToFront scr2&:POKEW &HDFF0*256+&H96,
      32:POKEL &HDFF0*256+&H144,0"
Un    PRINT#1,"WHILE PEEK(-4202495) AND 64"
x6    PRINT#1," CLS"
yg    IF flag3 THEN PRINT#1,"  Move rp&,0,251:Draw rp&,xf%,yf%
      :Draw rp&,631,251"
qQ    FOR ii=1 TO linie(0)STEP 2
```

Listing. Dreidimensionale Animationen mit »3DAnim« (Fortsetzung)

```
s54      a$=MID$(STR$(linie(ii+1)),2)
Eq       b$="Draw":IF linie(ii)=1 THEN b$="Move"
vp       PRINT#1,"  "+b$+" rp&,x%("+a$+",j%),y%("+a$+",j%)"
5A2    NEXT
Oi     Read.data
v8     DATA "  BltBitMap bm1&,0,0,bm2&,0,0,640,256,&HC0,-1,0"
LC     DATA "  j%=(j%+1)MOD(n%+1)"
wf     DATA WEND,WINDOW CLOSE 2,SCREEN CLOSE 1,WINDOW CLOSE 2,SC
       REEN CLOSE 2
Rh     DATA LIBRARY CLOSE,*
5Y     CLOSE
T50  RETURN
ks   '----- Assemblerprogramm auf Diskette erzeugen
HL   Assembler.code:
3c   Assembler.code.data:RESTORE Assembler.code.data
GW2    OPEN pname$+".asm"FOR OUTPUT AS 1
HG     PRINT#1," bra run"
mA     PRINT#1,"intname: dc.b 'intuition.library',0"
14     PRINT#1,"gfxname: dc.b 'graphics.library'"
35     PRINT#1,"title: dc.w 0"
1n     PRINT#1,"dosname: dc.b 'dos.library',0"
Gy     Read.data
Mu     DATA "defscr: dc.w 0,0,640,256,1,1,$8000,15"," dc.l 0,tit
       le,0,0"
pk     DATA "defwin: dc.w 0,0,640,256,1"," dc.l 0,$800,0,0,title
       "
95     DATA "screen1: dc.l 0,0"," dc.w 0,0,640,256,15","screen2:
        dc.l 0"
HH     DATA "window1: dc.l 0","rastport: dc.l 0","bitmap1: dc.l
       0"
P3     DATA "window2: dc.l 0","bitmap2: dc.l 0","adr: dc.l 0",*
Sn     PRINT#1,"filename: dc.b '"+pname$+".dat',0"
N5     Read.data
O1     DATA " even","filehd: dc.l 0","bild: dc.w 0","run: move.l
        4,a6",*
qZ     PRINT#1,"p: equ"+STR$(p+1)
XM     PRINT#1,"n: equ"+STR$(n+1)
Of     PRINT#1,"xf: equ "+STR$(xf)
40     PRINT#1,"yf: equ "+STR$(255-yf/2)
TB     Read.data
uy     DATA " lea dosname,a1"," jsr -408(a6)"," move.l d0,a3","
       move.l #filename,d1"
Gp     DATA " move.l #1005,d2"," jsr -30(a3)"," move.l d0,fileh
       d"," bne w2"
Ow     DATA " move.l a3,a1"," jmp -414(a6)","w2: move.l #n*p*4,
       d0"," moveq #2,d1"
Fx     DATA " jsr -198(a6)"," move.l d0,adr"," move.l d0,d2"," m
       ove.l filehd,d1"
pG     DATA " move.l #n*p*4,d3"," jsr -42(a3)"," move.l d0,d6",
       " move.l filehd,d1"
59     DATA " jsr -36(a3)"," move.l a3,a1"," jsr -414(a6)"
12     DATA " lea gfxname,a1"," jsr -408(a6)"," move.l d0,a3"
5W     DATA " lea intname,a1"," jsr -408(a6)"," move.l d0,a4"
NJ     DATA " move.l d0,a6"," lea defscr,a0"," jsr -198(a6)"
au     DATA " move.l d0,screen1"," lea defwin,a0"," jsr -204(a6)
       "
LN     DATA " move.l d0,window1"," move.l d0,a0"," move.l 50(a0)
       ,a0"
Ot     DATA " move.l a0,rastport"," move.l 4(a0),bitmap1"
tG     DATA " lea defscr,a0"," jsr -198(a6)"," move.l d0,screen2
       "
yH     DATA " add.l #184,d0"," move.l d0,bitmap2"
ab     DATA " sub.l #140,d0"," move.l d0,a0"," moveq #0,d0",*
wL     PRINT#1," moveq #"+MID$(STR$(ink(0)),2)+",d1"
7Y     PRINT#1," moveq #"+MID$(STR$(ink(1)),2)+",d2"
I1     PRINT#1," moveq #"+MID$(STR$(ink(2)),2)+",d3"
mU     Read.data
Ox     DATA " move.l a3,a6"," move.l a0,-(sp)"
Zp     DATA " jsr -288(a6)"," move.l (sp)+,a0"," moveq #1,d0",*
Hj     PRINT#1," moveq #"+MID$(STR$(ink(3)),2)+",d1"
Sw     PRINT#1," moveq #"+MID$(STR$(ink(4)),2)+",d2"
d9     PRINT#1," moveq #"+MID$(STR$(ink(5)),2)+",d3"
sa     Read.data
W6     DATA " jsr -288(a6)"," move #n,bild"," move #32,$dff096
       "," clr.l $dff144"
W1     DATA " move.l a3,a6"," move.l adr,a5","loop: move.l rastp
       ort,d7"
1c     DATA " move.l d7,a1"," moveq #0,d0"," moveq #0,d1"," js
       r -240(a6)"
R8     DATA " move.l d7,a1"," jsr -48(a6)",*
3h     IF flag3 THEN CALL Read.data
zP     DATA " move.l d7,a1"," moveq #0,d0"," move #255,d1"," j
       sr -240(a6)"
LE     DATA " move.l d7,a1"," move #xf,d0"," move #yf,d1"," js
       r -246(a6)"
GK     DATA " move.l d7,a1"," move #639,d0"," move #255,d1","
       jsr -246(a6)",*
wW     FOR ii=1 TO linie(0)STEP 2
wV4      PRINT#1," move.l d7,a1"
kC       a=4*linie(ii+1):IF a THEN a$=STR$(a)ELSE a$=" "
yK       PRINT#1," move "+a$+"(a5),d0"
x3       PRINT#1," move "+STR$(a+2)+"(a5),d1"
3G       PRINT#1," jsr "+STR$(-234-6*linie(ii))+"(a6)"
DI2    NEXT
sL0  datas1:RESTORE datas1
9r2    Read.data
OR     DATA " move.l bitmap1,a0"," moveq #0,d0"," moveq #0,d1"
Sz     DATA " move.l bitmap2,a1"," moveq #0,d2"," moveq #0,d3"
vQ     DATA " move #640,d4"," move #256,d5"," moveq #-$40,d6"
QT     DATA " moveq #-1,d7"," move.l d0,a2"," jsr -30(a6)"
6d     DATA " move bild,d0"," subq #1,d0"," bne w1"," move #n,
       d0"
JR     DATA " sub.l #n*p*4,a5","w1: move d0,bild"," add.l #p*4
       ,a5"
PJ     DATA " btst #6,$bfe001"," bne loop"
B1     DATA " move.l a4,a6"," move.l window1,a0"," jsr -72(a6)"
OA     DATA " move.l screen2,a0"," jsr -66(a6)"," move.l screen1
       ,a0"," jsr -66(a6)"
9n     DATA " move.l 4,a6"," move.l adr,a1"," move.l #n*p*4,d0"
       ," jsr -210(a6)"
ge     DATA " move.l a4,a1"," jsr -414(a6)"," move.l a3,a1"," jm
       p -414(a6)"," end",*
Lo     CLOSE
JL0  RETURN
1K   '***** Unterprogramme (SUB) :
LR   '----- x auf dem Stapel merken
rW   SUB Push(x)STATIC
602    stapel$=STR$(x)+CHR$(0)+stapel$
UW0  END SUB
7R   '----- x vom Stapel holen
1d   SUB Pop(x)STATIC
zJ2    x=VAL(stapel$):stapel$=MID$(stapel$,INSTR(stapel$,CHR$(0)
       )+1)
Ya0  END SUB
8H   '----- DATAs bis "*" lesen und auf Disk schreiben
sk   SUB Read.data STATIC
CP2    READ a$:WHILE a$<>"*":PRINT#1,a$:READ a$:WEND
ce0  END SUB
```

Listing 1. Dreidimensionale Animationen mit »3DAnim« (Schluß)

```
DIM 54,144

ORIG 50,100,0

SET 0,100,150:150,250:200,150:125,200:175,200                ;A
SET 5,210,150:210,250:240,220:270,250:270,150                ;M
SET 10,280,150:280,250                                       ;I
SET 12,350,250:290,250:290,150:350,150:350,200:320,200       ;G
SET 18,360,150:410,250:460,150:385,200:435,200               ;A
SET 23,10,20:10,120:40,90:70,120:70,20                       ;M
SET 28,80,20:130,120:180,20:105,70:155,70                    ;A
SET 33,240,120:190,120:190,20:240,20:240,70:210,70           ;G
SET 39,250,20:300,120:350,20:275,70:325,70                   ;A
SET 44,360,120:410,120:360,20:410,20                         ;Z
SET 48,420,20:420,120                                        ;I
SET 50,430,20:430,120:480,20:480,120                         ;N

LINE 0,1,2:3,4:5..9:10,11:12..17:18..20:21,22
LINE 23..27:28..30:31,32:33..38:39..41:42,43:44..47:48,49:50..53

MOVE 0..22:0,n/2-1,n-1,2ROT 300,100,0,300,100,10,360

ROT 300,100,0,300,100,10,360

MOVE 23..p-1:0,n/2-1,n-1,2

ROT 300,100,0,300,100,10,-360
```

Listing 2. Ein Beispiel für die Fähigkeiten von »3DAnim« liefern diese Befehle

Bitte mit dem Checksummer (Seite 143) eingeben. Der Checksummer und dieses Listing befinden sich auf der Programmservice-Diskette dieser Ausgabe (siehe auch Seite 34)

Nur Fliegen

Ein Flugsimulator zum Abtippen: Das gibt es nur im AMIGA-Magazin und im AMIGA-Sonderheft. Zumindest in dieser hervorragenden Geschwindigkeit, tollen Grafik und der Kürze des Listings.

von Michael Koch

Wollten Sie schon immer einmal fliegen? Die billigste und ungefährlichste Lösung ist ein Flugsimulator. Es ist schon erstaunlich, daß ein solches Programm in einer Hochsprache auf dem Amiga so kurz und leistungsfähig sein kann. Nicht nur die Landebahn, sondern Funktürme, ein Wald, ein See mit Insel, sich drehende Windräder und Vögel sind in »Flusi« realisiert. Erweiterungen um andere Landschaftselemente stellen für den geübten C-Programmierer kein Problem dar.

Nach dem Abtippen und Übersetzen starten Sie das Programm vom CLI aus mit folgendem Aufruf:

```
Flusi
```

Schnell ist der Bildschirmaufbau beendet, und Sie können losfliegen. Aber halt, erst ein paar Erläuterungen zur Steuerung. Die Instrumente sind im unteren Drittel angeordnet (Bild 1, Seite 41 oben; und Bild 2). Der Geschwindigkeitsmesser zeigt die Fluggeschwindigkeit in Knoten an. Bei zirka 50 Knoten können Sie abheben. Auf dem Kompaß sehen Sie die aktuelle Flugrichtung. Der Drehzahlmesser zeigt die Umdrehungszahl des Motors an. Die Instrumente Quer-, Seiten- und Höhenruder teilen Ihnen die aktuelle Stellung der Ruder mit. Achtung, sie schwenken nicht automatisch in die Neutralstellung zurück. Beim Höhenmesser ist der kleine Zeiger für die Zehner- und der große für die Hunderterstellen zuständig. Die Tausenderstelle wird durch einen Punkt dargestellt. Das Variometer zeigt die Steiggeschwindigkeit in 100 ft/Minute.

Um Ihr Flugzeug zu steuern, gibt es zwei Wege: mit dem Joystick in Port 2 oder über die Tastatur. In Bild 3 sehen Sie die Bedienung mit dem Joystick. Alle Funktionen sind allerdings nicht mit dem Joystick aufzurufen. Die Blickrichtung aus dem Cockpit läßt sich nur über die Tastatur ändern. Die Belegung sehen Sie in Bild 4. Die Bedeutung der restlichen Tasten, mit denen Sie das Flugzeug steuern können, finden Sie in der Tabelle. Auch während der Steuerung mit dem Joystick ist es möglich, die Tastatur zu benutzen.

Um Fliegen zu lernen, ist es am besten, wenn Sie es einfach ausprobieren. Bei eventuellen Abstürzen – des Flugzeugs und nicht des Computers – wird Ihnen nichts passieren, außer einem Farbflimmern des Bildschirms. Am besten ist die sanfte Methode: Also nicht zu heftig gegensteuern, wenn Sie die Flugrichtung ändern wollen, sonst kommt das Flugzeug schnell ins Trudeln.

Wenn Sie den Bogen heraus haben, können Sie noch viele Parameter für Ihren Flug verändern. Dazu drücken Sie während der Simulation <Esc>. In dem CLI-Fenster, von dem aus Sie Flusi gestartet haben, erscheint ein Menü. Damit Sie es ganz sehen, vergrößern Sie das Fenster vor dem Start auf volle Bildschirmgröße.

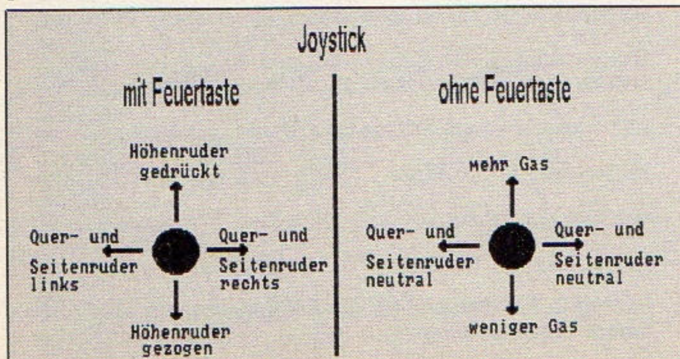


Bild 3. Die Steuerung von »Flusi« mit dem Joystick

1 bis 8

Mit diesen Menüpunkten können Sie verschiedene Grafikelemente ein- oder ausschalten. Das geschieht durch Eingabe der Menükennung und nachfolgendem <RETURN>. Zur nochmaligen Umschaltung wiederholen Sie die Eingabe.

K

Normalerweise sind Quer- und Seitenruder miteinander gekoppelt. Wenn Sie also eines davon bewegen, wird das andere automatisch verstellt. Ist die Kopplung ausgeschaltet, steuern Sie mit dem Joystick nur das Querruder. Das Seitenruder ist dann nur noch über die Tastatur zu beeinflussen.

4	6	Querruder
2	8	Höhenruder
3	9	Gashebel
0	.	Seitenruder
5		Seiten- und Querruder neutral
1		Bremse
Esc		Menü



Die Tasten, mit denen Sie »Flusi« steuern können

X, Y, Z

Diese drei Parameter bestimmen Ihre Position. Die Grenzwerte stehen hinter der Bezeichnung in Klammern. Geben Sie Werte außerhalb dieser Grenzen ein, verlangt Flusi die erneute Eingabe.

S

Der Steigungswinkel ist der Winkel, in dem Ihr Flugzeug im Verhältnis zur Erde fliegt. 90 Grad entspricht einem Flug senkrecht nach oben. Bei -90 Grad fliegen Sie direkt auf den Erdboden zu. Für einen waagrechten Flug geben Sie als Wert 0 ein.

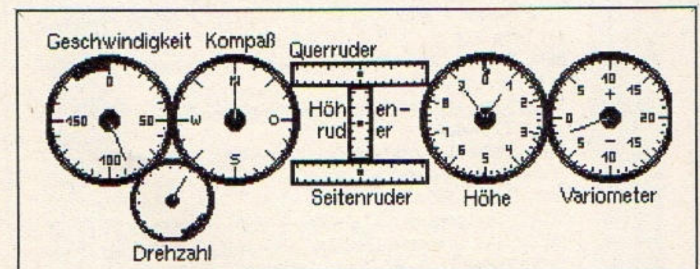


Bild 2. Die Instrumente von »Flusi« mit Erklärung

N

Mit diesem Menüpunkt bestimmen Sie die Himmelsrichtung, in die Sie fliegen wollen. Folgende Werte entsprechen den vier Himmelsrichtungen (alle Werte dazwischen sind erlaubt):

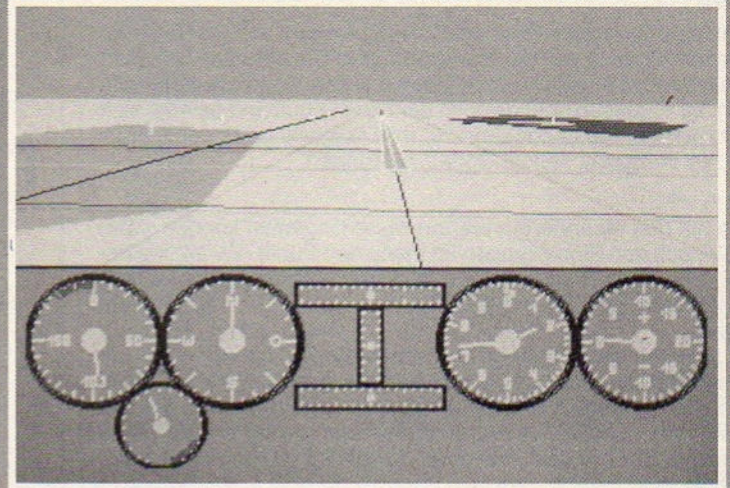
0°	Nord
90°	Ost
-180°, 180°	Süd
-90°	West

Michael Koch



Die »Computerkarriere« von Michael Koch begann im Jahr 1983 auf einem Sharp PC-1401 in Basic. Doch dabei blieb es nicht. 1985 folgte ein Apple II, auf dem er nach Basic auch in 6502 Assembler programmierte. Seinen Amiga 1000 kaufte er 1987. Auch hier begann er mit Basic und Assembler, stieg dann aber auf C um. Seine Hobbies sind Astronomie, Elektronik und natürlich der Computer.

ist schöner



Bildverweis: Bavaria

H

Der Hängewinkel entspricht der Lage Ihres Flugzeuges. Positive Werte entsprechen einer Schräglage nach rechts, negative einer Lage nach links. Bei 180 und -180 Grad »liegt« Ihr Flugzeug auf dem Rücken.

G, L, B

Diese Werte sind für Fluggeschwindigkeit (G) und Landebahnabmessungen (Länge, Breite) zuständig.

U, I

Dies sind zwei feste Positionen. Nach Anwahl von »U« befindet sich das Flugzeug auf der Startbahn, und Sie können vom Start weg alles selbst erledigen. Zum Üben des Landeanflugs geben Sie »I« ein. Ihr Flugzeug befindet sich dann in der Luft und bewegt sich fast korrekt auf die Landebahn zu. Ihre Aufgabe ist es dann »nur« noch, Ihr Fluggerät heil zu Boden zu bringen.

W, Q

Um mit den veränderten Werten weiterfliegen zu können, wählen Sie nach den Einstellungen im Menü den Punkt »W« an. Sie befinden sich dann wieder im Flugzeug und können weitermachen. Zum Verlassen von Flusi geben Sie <q> ein. Das Programm endet dann.

Flusi ist multitaskingfähig. Da jedoch die Bildschirme des Programms ständig aktiviert und nach vorne gebracht werden, sind Eingaben in anderen Fenstern unmöglich. Ein Programm, das keine Eingaben benötigt, kann aber im Hintergrund laufen. Das funktioniert gut, da Flusi beim Fliegen nur ungefähr 160 KByte Speicher benötigt.

Ihr Flugzeug verträgt auch einen größeren Motor, sprich eine 68020-Karte. Die Geschwindigkeit beim Fliegen steigert sich dann noch einmal. Natürlich ist Flusi keine perfekte Simulation, aber es zeigt deutlich, was mit dem Amiga möglich ist. Die wichtigsten Eigenschaften sind berücksichtigt und machen diese Simulation zu einem Programm, das man sicher oft und lange spielen wird. Guten Flug.

René Beaupoil/kn

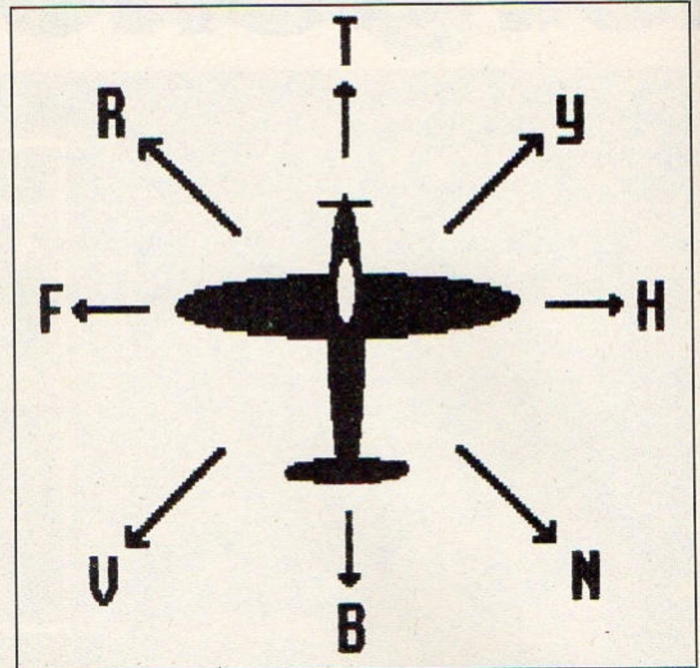


Bild 4. Die verschiedenen Blickrichtungen aus dem Flugzeug und die entsprechenden Tasten

Programmname:	Flusi
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.4, V3.6
Aufrufe:	CC Flusi LN Flusi.o -LM -LC

```
Programmautor: Michael Koch
---------------------------
/* Flugsimulator F L U S I 1 . 0 Michael Koch  März 1989 *
/
#include <exec/types.h>
#include <exec/libraries.h>
#include <intuition/intuition.h>
#include <intuition/intuitionbase.h>
#include <stdio.h>
#define CLEAR printf("%c",12)
#define PF printf
struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
struct Library *OpenLibrary();
struct Screen *screen2,*screen3,*screen4,*OpenScreen();
struct Window *window1,*window2,*window3,*window4,*OpenWin
dow();
struct RastPort *rp,*rp2,*rp3,*rp4;
struct IntuiMessage *nachricht,*GetMsg();
struct NewScreen newscreen =
 { 0,0,320,160,3,0,1,NULL,CUSTOMSCREEN,NULL,NULL,NULL,NULL
  };
struct NewWindow newwindow =
 { 0,0,320,160,0,1,VANILLAKEY,BORDERLESS | NOCAREREFRESH,
   NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CUSTOMSCRE
   EN };
struct NewScreen newscreen4 =
 { 0,163,320,96,2,0,1,NULL,CUSTOMSCREEN,NULL,NULL,NULL,NUL
  L };
struct NewWindow newwindow4 =
 { 0,0,320,96,0,1,NULL,BORDERLESS | NOCAREREFRESH,
   NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CUSTOMSCRE
   EN };
struct AreaInfo a_info;
struct TmpRas t_ras;
USHORT code;
UWORD areabuf[250],*AllocMem(),*maus;
BYTE  *area_plane,*AllocRaster();
char  input[40];
float ph,pi,zp,t01,t02,t03,t04;
float st,ha,az,cst,sbl,sha,cha,cbl,sst,saz,caz;
float pit,nenner,hea,ban,siban,coban,sihea,cohea,sipit,cop
it;
float absi,abco,winkel,weg,geschw,altgeschw,vario;
float sin(),cos(),tan(),atan(),sqrt();
int   knueppel,taste,c,fenster,det = 0xffff,ende = 0;
int   t1,t2,t3,t4,t5,t6,t7,t8,t9,dx1,dy1,dz1,dx2,dy2,dz2;
int   projx1,projx2,projy1,projy2,punktx,punkty,pktx,pkty;
int   hruder,qruder,sruder,upm,gas,flag,flag1,flag2;
int   bahnx1,bahnx2,bahnx3,bahnx4,bahny1,bahny2,bahny3,bah
ny4;
int   bahnx5,bahnx6,bahnx7,bahnx8,bahny5,bahny6,bahny7,bah
ny8;
long  transx1,transy1,transz1,transx2,transy2,transz2;
long  tx1,tx2,ty1,ty2,tz1,tz2,txalt2,tyalt2,tzalt2,k;
long  flugx,flugy,flugz,altflugz,pot,abst,abst2,count = 0;
long  x1,x2,x3,x4,x5,x6,y1,y2,y3,y4,y5,y6,z1,z2;
long  deltax1,deltay1,deltaz1,deltax2,deltay2,deltaz2;
long  zeiger0x =  64, zeiger0y = 75, zeiger1x = 224, zeige
r1y = 34;
long  zeiger2x = 224, zeiger2y = 34, zeiger3x =  -5, zeige
r3y = -5;
long  zeiger4x =  32, zeiger4y = 34, zeiger5x = 288, zeige
r5y = 34;
long  zeiger6x =  96, zeiger6y = 34;
long  zeiger7x =  -5, zeiger8y = -5, zeiger9x = -5;
long  vogel,vogelx,vogely,vogelz,bahnlaenge,bahnbreite;
long  mikrosek,altmikrosek,sek,altsek,zeitdiff,atol();
int   ziffer[10] = {0x7b6f,0x1749,0x73e7,0x73cf,0x5bc9,
                    0x79cf,0x79ef,0x7249,0x7bef,0x7bcf};
long  zeichen[4] = {0x47359c40,0x3a318b80,0x3e0e0f80,0x463
5aa80};
flugmechanik()
{
  /* Folgende globale Variablen stehen zur Verfügung :
     int   hruder,sruder,qruder   Ruderstellungen (-128...
     128)
     int   gas                         Gashebel-Stellung in Pro
     zent (0...100)
     long  zeitdiff          Zeit seit der letzten Berechnun
     g in Mikrosec.
```

```
    Folgende globale Variablen stehen zur Verfuegung und
    muessen
     in flugmechanik() neu berechnet werden :
    long  flugx,flugy,flugz        Koordinaten des Flugzeug
    s in cm
    float st                       Steigungswinkel (-PI/2..
    .PI/2)
    float az                       Azimutwinkel    (-PI...P
    I)
    float ha                       Haengewinkel    (-PI...P
    I)
    int   upm                      Motor-Drehzahl in 1/min
    (0...4000)
    float geschw                   Geschwindigkeit in Knote
    n
    Es ist wichtig, dass die Winkel sich innerhalb der an
    gegebenen
    Bereiche befinden ! */
  st -= 0.000015 * hruder * cha * geschw;  /* Normalflug
  und Rueckenflug */
  st -= 0.000005 * sruder * sha * geschw;  /* Steilkurven
   */
  st += (geschw - 90) * 0.0003;    /* Geschwindigkeitsabha
  engigkeit */
  if (geschw < 50 && flugz > 202) st -= (50 - geschw) *
  0.01;  /* Stall */
  if (st > ph || st < -ph){              /* Looping-Korre
  ktur */
    st = pi - st; if (st > ph) st -= zp;
    az += pi; if (az > pi) az -= zp;
    ha += pi; if (ha > pi) ha -= zp;
  }
  ha += 0.0006 * qruder;           /* Einfluss des Querrud
  ers */
  ha += 0.0003 * sruder;           /* Einfluss des Seitenr
  uders */
  if (ha> pi) ha -= zp;
  if (ha<-pi) ha += zp;
  if (ha < 1 && ha > -1) ha *= 0.995;      /* --> stabi
  le Fluglage */
  az += 0.000001 * sruder * cha * geschw;  /* Einfluss de
  s Seitenruders */
  az += 0.000001 * hruder * sha * geschw;  /* Einfluss de
  s Hoehenruders */
  if (flugz < 202) az += 0.0001 * sruder;  /* Bugrad-Len
  kung */
  if (ha < ph && ha > -ph) az += 0.0004 * ha * geschw; /
  * Haengew.-Einfl. */
  while (az > pi) az -= zp;
  while (az <-pi) az += zp;
  sst = sin(st); cst = cos(st); saz = sin(az); caz = cos(a
  z);
  sha = sin(ha); cha = cos(ha);       /* wichtig fuer weite
  re Berechnungen */
  upm = 500 + gas * 20;                          /* Drehza
  hl in 1/min */
  if (geschw > 100) upm += (geschw - 100) * 20;  /* erhoe
  hte Drehzahl */
  altgeschw = geschw;
  geschw = 20 + upm * .04 - 200 * sst;     /* Geschwindigk
  eit in Knoten */
  if (flugz <= 200 && gas == 0) geschw = 0;  /* ausrollen
   */
  geschw = altgeschw + (geschw - altgeschw) / 175;
  weg = zeitdiff * geschw * .0000514;      /* Weg in cm */
  flugx += weg * cst * saz;                /* neue Koordin
  aten in cm */
  flugy += weg * cst * caz;
  flugz += weg * sst;
  if (flugz < 202){
    flugz = 200;                  /* tiefer geht es nicht !
    */
    if (st < -0.1 || ha > 0.1 || ha < -0.1) crash();
    ha = 0; st = 0;               /* Bodenkontakt */
  }
}
ruder()
{
#asm
  move.w $dff00c,_knueppel                  /* Joystick-Abfrag
  e */
  move.w $bfe000,_taste
#endasm
  if (taste & 0x0080){
    if (knueppel & 2) qruder += 8;           /* Ruderstellu
    ngen aendern */
    if (qruder > 128) qruder = 128;
    if (knueppel & 512) qruder -= 8;
    if (qruder < -128) qruder = -128;
    if (((knueppel >> 1) ^ knueppel) & 1) hruder -= 4;
    if (hruder < -128) hruder = -128;
    if (((knueppel >> 1) ^ knueppel) & 256) hruder += 4;
    if (hruder > 128) hruder = 128;
    if (det & 0x0100) sruder = qruder;
  }
  else{
    if (((knueppel >> 1) ^ knueppel) & 1) gas -= 4;    /
    * Gas aendern */
    if (gas < 0) gas = 0;
    if (((knueppel >> 1) ^ knueppel) & 256) gas += 4;
    if (gas > 100) gas = 100;
    if (knueppel & 514) qruder = sruder = 0; /* Ruder in N
    eutralstellung */
  }
}
landschaft()
{
  pit = cst * sbl * sha - cbl * sst;                 /* Pit
  ch */
  if (pit >=  1.0 || pit <= -1.0)
    (pit > 0) ? (pit = -ph) : (pit = ph);
  else           /* arcsin */
    pit = -atan(pit / sqrt(1.0 - pit * pit));
  hea = cha * sbl;                                   /* Hea
  ding */
  nenner = cbl * cst - sbl * sst * sha;
  if (nenner != 0){
    hea = az + atan(hea / nenner);
    if (nenner < 0) hea += pi;
  }
  else{
    if (cha > 0)
      (sbl > 0) ? (hea = az + ph) : (hea = az - ph);
    else
      (sbl > 0) ? (hea = az - ph) : (hea = az + ph);
  }
  while (hea > pi) hea -= zp;
  if (hea <-pi) hea += zp;
  ban = cbl * sha + sbl * sst;                       /* Ban
  k */
  nenner = cha * cst;
  if (nenner == 0)
    (ban < 0) ? (ban = ph) : (ban = -ph);
  else{
    ban = -atan(ban / nenner);
    if (nenner < 0)
    { ban += pi; if (ban > pi) ban -= zp; }
  }
  siban = sin(ban); coban = cos(ban);
  sihea = sin(hea); cohea = cos(hea);
  sipit = sin(pit); copit = cos(pit);
  t01 =  cohea * coban; t02 = -sihea * coban;  /* Transfor
  mationsmatrix */
  t03 = -cohea * siban; t04 =  sihea * siban;
  t1 = (int) (4096 * (t01 - t04 * sipit));
  t2 = (int) (4096 * (t02 + t03 * sipit));
  t3 = (int) (4096 * (copit * siban));
  t4 = (int) (8192 * (t03 + t02 * sipit));
  t5 = (int) (8192 * (t04 - t01 * sipit));
  t6 = (int) (8192 * (copit * coban));
  t7 = (int) (4096 * (sihea * copit));
  t8 = (int) (4096 * (cohea * copit));
  t9 = (int) (4096 * (sipit));
  flag = flag1 = flag2 = 0;      /* Horizont in 5000km Ent
  fernung */
  x3 =  500000000 * sihea; y3 =  500000000 * cohea;
  x4 = y3 << 1; y4 = x3 << 1;
  x1 = x3 - x4; y1 = y3 + y4; x2 = x3 + x4; y2 = y3 - y4;
  z1 = z2 = 0;
  kante(0);
  if (flag){
    SetAPen(rp,7L);
    RectFill(rp,0L,0L,319L,159L);       /* altes Bild loes
```

Listing. Ein toller Flugsimulator mit allem Drum und Dran: »Flusi«. Bitte mit dem Checksummer (Seite 143) eingeben.

```
chen */
    SetAPen(rp,1L);
    projx1 = pktx; projy1 = pkty; ecke();       /* Land fuellen */
    AreaEnd(rp);
  }
  else{
    (pit>0) ? (SetAPen(rp,7L)) : (SetAPen(rp,1L));
    RectFill(rp,0L,0L,319L,159L); /* alles fuellen */
  }
  if (det & 0x08){ /* Stadt */
    flag = flag1 = flag2 = 0;
    z1 = z2 = 0;
    x1 = -200000; y1 = -200000; x2 = -100000; y2 = -600000; kante(0);
    x1 = -100000; y1 = -600000; x2 = -600000; y2 = -900000; kante(1);
    x1 = -600000; y1 = -900000; x2 = -500000; y2 = -100000; kante(1);
    x1 = -500000; y1 = -100000; x2 = -200000; y2 = -200000; kante(1);
    if (flag){
      projx1 = pktx; projy1 = pkty; ecke();
      SetAPen(rp,2L); AreaEnd(rp);
    }
  }
  if ((det & 0x01) && (pit < 0.7)){ /* Erdboden mit Gitter versehen */
    (sihea > 0) ? (absi = sihea) : (absi = -sihea);
    (cohea > 0) ? (abco = cohea) : (abco = -cohea);
    x3 = x5 = flugx + flugz * ( 0.7 * sipit + copit * (4 * sihea - 2 - abco));
    x4 = x6 = flugx + flugz * (-0.7 * sipit + copit * (4 * sihea + 2 + abco));
    y3 = y5 = flugy + flugz * ( 0.7 * sipit + copit * (4 * cohea - 2 - absi));
    y4 = y6 = flugy + flugz * (-0.7 * sipit + copit * (4 * cohea + 2 + absi));
    for (pot = 12500; pot < 100000000; pot <<= 1){
      if (flugz < (pot - (pot >> 2))){
        abst  = pot << 1; abst2 = abst << 3;
        if (flugz > (pot >> 1))
          SetAPen(rp,4L);
        else
          SetAPen(rp,0L);
        z1 = z2 = 0;
        x3 -= (x3 - pot) % abst; x4 -= (x4 + pot) % abst;
        y3 -= (y3 - pot) % abst; y4 -= (y4 + pot) % abst;
        gitter();
        SetAPen(rp,0L);
        x3 = x5 - x5 % abst; x4 = x6 - x6 % abst;
        y3 = y5 - y5 % abst; y4 = y6 - y6 % abst;
        gitter();
        break;
      }
    }
  }
#define MAKRO5 {ecke();
\
                if (flag)
                \
                  AreaDraw(rp,(long)projx1,(long)projy1);
                  \
                else
                \
                  AreaMove(rp,(long)projx1,(long)projy1);
                  \
                AreaDraw(rp,(long)projx2,(long)projy2);
                }
  if (det & 0x02){ /* Startbahn */
    flag = flag1 = flag2 = 0;
    z1 = z2 = 0;
    x1 = -bahnbreite; y1 = -bahnlaenge; x2 = -bahnbreite; y2 =  bahnlaenge;
    if (projektion(0)){
      bahnx1=projx1; bahny1=projy1; bahnx2=projx2; bahny2=projy2;
      MAKRO5; flag += 1;
    }
    x2 =  bahnbreite; y2 = bahnlaenge;
    if (projektion(1)){
      bahnx3=projx1; bahny3=projy1; bahnx4=projx2; bahny4=projy2;
      MAKRO5; flag += 2;
    }
    x2 = bahnbreite; y2 = -bahnlaenge;
    if (projektion(1)){
      bahnx5=projx1; bahny5=projy1; bahnx6=projx2; bahny6=projy2;
      MAKRO5; flag += 4;
    }
    x2 = -bahnbreite; y2 = -bahnlaenge;
    if (projektion(1)){
      bahnx7=projx1; bahny7=projy1; bahnx8=projx2; bahny8=projy2;
      MAKRO5; flag += 8;
    }
    if (flag){
      projx1 = pktx; projy1 = pkty; ecke();
      SetAPen(rp,2L);                        /* graue Startbahn */
      AreaEnd(rp);
    }
    SetAPen(rp,5L);              /* Weisse Linien */
    if (flag & 1) {
      Move(rp,(long)bahnx1,(long)bahny1); Draw(rp,(long)bahnx2,(long)bahny2);
    }
    if (flag & 2){
      Move(rp,(long)bahnx3,(long)bahny3); Draw(rp,(long)bahnx4,(long)bahny4);
    }
    if (flag & 4){
      Move(rp,(long)bahnx5,(long)bahny5); Draw(rp,(long)bahnx6,(long)bahny6);
    }
    if (flag & 8){
      Move(rp,(long)bahnx7,(long)bahny7); Draw(rp,(long)bahnx8,(long)bahny8);
    }
    if (flugz < 100000 && flugx > -30000 && flugx < 30000){
      x1 = x2 = 0;
      /* Mittellinie */
      if (flugz > 20000){ /* Flughoehe ueber 200m */
        y1=-bahnlaenge; y2= bahnlaenge; linie();
      }
      else{                    /* Flughoehe unter 200m */
        y3 = flugy - (flugz << 2) - 5000;
        y4 = flugy + (flugz << 2) + 15000;
        y3 -= y3 % 10000; y4 -= y4 % 10000;
        if ((y4 <= -bahnlaenge) || (y3 >= bahnlaenge)){
          y1=-bahnlaenge; y2= bahnlaenge; linie();
        }
        else{
          if (y3 <= -bahnlaenge) y3 = -bahnlaenge;
          else{
            y1 = -bahnlaenge;       /* durchgezogene Linie am neg. Ende */
            y2 = y3 - 3000; linie();
          }
          if (y4 >= bahnlaenge) y4 = bahnlaenge;
          else{
            y1 = bahnlaenge;        /* durchgezogene Linie am pos. Ende */
            y2 = y4; linie();
          }
          for (y1=y3; y1<y4; y1+=10000){
            y2 = y1+7000; linie();        /* die Striche sind 70m lang */
          }
        }
      }
    }
  }
  if (det & 0x04){                          /* See */
    flag = flag1 = flag2 = 0;
    z1 = z2 = 0;
    x1 = 200000; y1 = 300000; x2 = 500000; y2 = 600000; kante(0);
    x1 = 500000; y1 = 600000; x2 = 800000; y2 = 200000; kante(1);
    x1 = 800000; y1 = 200000; x2 = 400000; y2 =-200000; kante(1);
    x1 = 400000; y1 =-200000; x2 = 200000; y2 = 300000; kante(1);
    if (flag){
      projx1 = pktx; projy1 = pkty; ecke();
```

```
      SetAPen(rp,6L); AreaEnd(rp);
    }
    flag = flag1 = flag2 = 0;              /* Insel */
    z1 = z2 = 0;
    x1 = 350000; y1 = 250000; x2 = 500000; y2 = 250000; kante(0);
    x1 = 500000; y1 = 250000; x2 = 450000; y2 = 100000; kante(1);
    x1 = 450000; y1 = 100000; x2 = 350000; y2 = 250000; kante(1);
    if (flag){
      projx1 = pktx;projy1 = pkty;ecke();
      SetAPen(rp,1L); AreaEnd(rp);
    }
  }
  if (det & 0x20){
    x3 =  350000; y3 = -300000; windrad();
    x3 =  400000; y3 = -300000; windrad();
    x3 =  450000; y3 = -250000; windrad();
    x3 = -200000; y3 =  200000; windrad();
  }
  if (det & 0x10){
    x1 = -200000; y1 =  500000; funkturm();
    x1 = -350000; y1 =  250000; funkturm();
    x1 = -400000; y1 =  100000; funkturm();
    x1 = -350000; y1 = -250000; funkturm();
    x1 = -250000; y1 = -550000; funkturm();
    x1 = -450000; y1 = -650000; funkturm();
    x1 =  650000; y1 = -500000; funkturm();
    x1 =  450000; y1 =  200000; funkturm();
  }
  if (det & 0x40){                       /* Wald mit 10 Baeumen */
    SetAPen(rp,0L);
    x1 = 204000; y1 = 112000; baum(); x1 = 210000; y1 = 112000; baum();
    x1 = 190000; y1 = 110000; baum(); x1 = 198000; y1 = 108000; baum();
    x1 = 206000; y1 = 104000; baum(); x1 = 196000; y1 = 102000; baum();
    x1 = 200000; y1 = 100000; baum(); x1 = 192000; y1 =  96000; baum();
    x1 = 204000; y1 =  94000; baum(); x1 = 196000; y1 =  90000; baum();
  }
  if (det & 0x80){                       /* Vogel */
    if (vogel){
      if ((vogel++) == 35) vogel = 0;    /* Der Vogel ist weg */
      else{
        SetAPen(rp,0L);
        x1 = vogelx; x2 = vogelx + 50; y1 = y2 = vogely;
        z1 = vogelz; (vogel%2) ? (z2 = z1 + 20) : (z2 = z1 - 20);
        linie(); x2 = vogelx - 50; linie();
        vogely -= 300;              /* Eigenbewegung des Vogels */
      }
    }
    else{
      if ((flugy & 511L) == 511L && flugz < 100000){   /* neuer Vogel ? */
        vogel = 1;
        vogelx = flugx + saz * 20000 + (flugx & 8191);
        vogely = flugy + caz * 20000 + (flugy & 8191);
        vogelz = flugz + 500 + (flugy & 1023);
        if (vogelz < 200) vogelz = 200;
      }
    }
  }
  x1 = x2 = 0; z1 = z2 = 500000000; y1 = y2 = -z2;   /* Sonne im Sueden */
  SetAPen(rp,5L);
  if (projektion(0))
    RectFill(rp,(long)projx1-1,(long)projy1-2,(long)projx1+1,(long)projy1);
  /******  Platz fuer Landschafts-Erweiterungen  ******/
}
kante(modus)
int modus;
{
  if (projektion(modus)){
    ecke();
    if (flag) AreaDraw(rp,(long)projx1,(long)projy1);
    else      AreaMove(rp,(long)projx1,(long)projy1);
    AreaDraw(rp,(long)projx2,(long)projy2);
    flag = 1;
  }
}
ecke()     /* eventuelles Auffuellen der Ecken beim Flaechenfuellen */
{
  if (flag2){        /* Quellpunkt ist schon vorhanden */
    if (punktx == 0){
      if (projy1 == 0) AreaDraw(rp,0L,0L);
      if (projx1 == 319)
      { AreaDraw(rp,0L,0L);AreaDraw(rp,319L,0L); }
      if (projy1 == 159)
      { AreaDraw(rp,0L,0L);AreaDraw(rp,319L,0L);AreaDraw(rp,319L,159L); }
    }
    if (punkty == 0){
      if (projx1 == 319) AreaDraw(rp,319L,0L);
      if (projy1 == 159)
      { AreaDraw(rp,319L,0L);AreaDraw(rp,319L,159L); }
      if (projx1 == 0)
      { AreaDraw(rp,319L,0L);AreaDraw(rp,319L,159L);AreaDraw(rp,0L,159L); }
    }
    if (punktx == 319){
      if (projy1 == 159) AreaDraw(rp,319L,159L);
      if (projx1 == 0)
      { AreaDraw(rp,319L,159L);AreaDraw(rp,0L,159L); }
      if (projy1 == 0)
      { AreaDraw(rp,319L,159L);AreaDraw(rp,0L,159L);AreaDraw(rp,0L,0L); }
    }
    if (punkty == 159){
      if (projx1 == 0) AreaDraw(rp,0L,159L);
      if (projy1 == 0)
      { AreaDraw(rp,0L,159L);AreaDraw(rp,0L,0L); }
      if (projx1 == 319)
      { AreaDraw(rp,0L,159L);AreaDraw(rp,0L,0L);AreaDraw(rp,319L,0L); }
    }
    flag2 = 0;   /* jetzt ist kein Quellpunkt mehr da */
  }
  if ((projx2==0) || (projx2==319) || (projy2==0) || (projy2==159)){
    flag2 = 1;   /* jetzt ist wieder ein Quellpunkt da */
    punktx = projx2; punkty = projy2;
  }
  if (flag1 == 0){  /* falls kein Zielpunkt da ist */
    if ((projx1==0) || (projx1==319) || (projy1==0) || (projy1==159)){
      flag1 = 1;   /* jetzt ist ein Zielpunkt da */
      pktx = projx1; pkty = projy1;
    }
  }
}
gitter()
{
  if (x3 > flugx) x3 -= abst;
  if (x4 < flugx) x4 += abst;
  if (y3 > flugy) y3 -= abst;
  if (y4 < flugy) y4 += abst;
  x1 = x3 - abst2; x2 = x4 + abst2;
  for (y1 = y3; y1 <= y4; y1 += abst)
    { y2 = y1; linie(); }
  y1 = y3 - abst2; y2 = y4 + abst2;
  for (x1 = x3; x1 <= x4; x1 += abst)
    { x2 = x1; linie(); }
}
baum()
{
  z1 = 0; z2 = 2000; x2 = x1; y2 = y1; linie();
  if (flugx > x1-40000 && flugx < x1+40000 && flugy > y1-40000 \
      && flugy < y1+40000 && flugz < 40000){
    z1 += 500; z2 -= 500; x2 = x1 + 500; y2 = y1 + 300; linie();
    z1 += 300;            x2 = x1;       y2 = y1 - 500; linie();
    z1 += 300;            x2 = x1 - 400; y2 = y1 + 200; linie();
  }
```

Listing. Ein toller Flugsimulator: »Flusi« (Fortsetzung)

```
}
funkturm()
{
 SetAPen(rp,5L);
 z1 = 0; z2 = 14000; x2 = x1; y2 = y1; linie();
 if (flugx > x1-100000 && flugx < x1+100000 && flugy >
  y1-100000 \
    && flugy < y1+100000 && flugz < 100000){
   SetAPen(rp,3L);
   for (z1=1500; z1<11000; z1+=3000)      /* rote Streife
   n */
       { z2 = z1 + 1500; linie(); }
 }
}
windrad()
{
 SetAPen(rp,5L);
 switch (count % 3){
   case 0 : x1 = x3 + 2000; x2 = x3 - 2000;         /* Flue
   gel */
            y1 = y3 + 2000; y2 = y3 - 2000;
            z1 = 3000; z2 = 7000; linie(); break;
   case 1 : x1 = x3 + 2732; x2 = x3 - 2732;
            y1 = y3 + 2732; y2 = y3 - 2732;
            z1 = 5732; z2 = 4268; linie(); break;
   case 2 : x1 = x3 +  732; x2 = x3 -  732;
            y1 = y3 +  732; y2 = y3 -  732;
            z1 = 7732; z2 = 2268; linie(); break;
 }
 if (flugx > x1-300000 && flugx < x1+300000 && flugy >
  y1-300000 \
    && flugy < y1+300000 && flugz < 300000){
   SetAPen(rp,3L);                                   /* T
   urm */
   z1 = 0; x2 = x3 + 500; y2 = y3 - 500; z2 = 5000;
   x1 = x3 + 1000; y1 = y3; linie();
   y1 = y3 - 1000; linie();
   x1 = x3; y1 = y3 - 1000; linie();
   x1 = x3; y1 = y3; linie();
   z1 = 5000; x2 = x3 + 1000; y2 = y3 - 1000; linie();
 }
}
linie()
{
 if (projektion(0)){
   Move(rp,(long)projx1,(long)projy1);
   Draw(rp,(long)projx2,(long)projy2);
 }
}
projektion(modus)
int modus;
{
 if (modus == 1)
 { tx1 = txalt2; ty1 = tyalt2; tz1 = tzalt2; }
 else{
   deltax1 = x1 - flugx; deltay1 = y1 - flugy; deltaz1 =
   z1 - flugz;
#define MAKRO1 { deltax1 >>= 2; deltay1 >>= 2; deltaz
1 >>= 2; }
   while (deltax1 > 32767 || deltax1 < -32768) MAKRO1;
   while (deltay1 > 32767 || deltay1 < -32768) MAKRO1;
   while (deltaz1 > 32767 || deltaz1 < -32768) MAKRO1;
   dx1 = (int)deltax1; dy1 = (int)deltay1; dz1 = (int)del
   taz1;
   transx1 = dx1 * (long)t1 + dy1 * (long)t2 + dz1 * (lon
   g)t3;
   transy1 = dx1 * (long)t4 + dy1 * (long)t5 + dz1 * (lon
   g)t6;
   transz1 = dx1 * (long)t7 + dy1 * (long)t8 + dz1 * (lon
   g)t9;
#define MAKRO2 { transx1 >>= 2; transy1 >>= 2; transz
1 >>= 2; }
   while (transx1 > 32767 || transx1 < -32768) MAKRO2;
   while (transy1 > 32767 || transy1 < -32768) MAKRO2;
   while (transz1 > 32767 || transz1 < -32768) MAKRO2;
   tx1 = (int)transx1; ty1 = (int)transy1; tz1 = (int)tra
   nsz1;
 }
 deltax2 = x2 - flugx; deltay2 = y2 - flugy; deltaz2 = z2
  - flugz;
#define MAKRO3 { deltax2 >>= 2; deltay2 >>= 2; deltaz
2 >>= 2; }
 while (deltax2 > 32767 || deltax2 < -32768) MAKRO3;
 while (deltay2 > 32767 || deltay2 < -32768) MAKRO3;
 while (deltaz2 > 32767 || deltaz2 < -32768) MAKRO3;
 dx2 = (int)deltax2; dy2 = (int)deltay2; dz2 = (int)delta
 z2;
 transx2 = dx2 * (long)t1 + dy2 * (long)t2 + dz2 * (long)
 t3;
 transy2 = dx2 * (long)t4 + dy2 * (long)t5 + dz2 * (long)
 t6;
 transz2 = dx2 * (long)t7 + dy2 * (long)t8 + dz2 * (long)
 t9;
#define MAKRO4 { transx2 >>= 2; transy2 >>= 2; transz
2 >>= 2; }
 while (transx2 > 32767 || transx2 < -32768) MAKRO4;
 while (transy2 > 32767 || transy2 < -32768) MAKRO4;
 while (transz2 > 32767 || transz2 < -32768) MAKRO4;
 tx2 = (int)transx2; ty2 = (int)transy2; tz2 = (int)trans
 z2;
 txalt2 = tx2; tyalt2 = ty2; tzalt2 = tz2;
 for (;;){
   c = 0;
   if (tx1 < -tz1) c+=  1; if (tx1 > tz1) c+=  2;
   if (ty1 < -tz1) c+=  4; if (ty1 > tz1) c+=  8;
   if (tx2 < -tz2) c+= 16; if (tx2 > tz2) c+= 32;
   if (ty2 < -tz2) c+= 64; if (ty2 > tz2) c+=128;
   if (c == 0){
     if (tz1 == 0) tz1 = 1;                       /* 3D --
     > 2D */
     if (tz2 == 0) tz2 = 1;
     projx1 = (319 + (tx1 * 319) / tz1) >> 1;
     projy1 = (159 - (ty1 * 159) / tz1) >> 1;
     projx2 = (319 + (tx2 * 319) / tz2) >> 1;
     projy2 = (159 - (ty2 * 159) / tz2) >> 1;
     return(1);
     break;
   }
   if (((c&17)==17) || ((c&34)==34) || ((c&68)==68) || ((
   c&136)==136)){
     return(0); break;     /* Linie ist nicht sichtbar */
   }
   if (c & 1){
     k   = ((tz1+tx1) << 15 )/(tx1-tx2-tz2+tz1);
     tx1 = (k * (tz1-tz2) >> 15) - tz1;
     ty1 = (k * (ty2-ty1) >> 15) + ty1;
     tz1 = -tx1; continue;
   }
   if (c & 2){
     k   = ((tz1-tx1) << 15 )/(tx2-tx1-tz2+tz1);
     tx1 = (k * (tz2-tz1) >> 15) + tz1;
     ty1 = (k * (ty2-ty1) >> 15) + ty1;
     tz1 = tx1; continue;
   }
   if (c & 4){
     k   = ((tz1+ty1) << 15 )/(ty1-ty2-tz2+tz1);
     tx1 = (k * (tx2-tx1) >> 15) + tx1;
     ty1 = (k * (tz1-tz2) >> 15) - tz1;
     tz1 = -ty1; continue;
   }
   if (c & 8){
     k   = ((tz1-ty1) << 15 )/(ty2-ty1-tz2+tz1);
     tx1 = (k * (tx2-tx1) >> 15) + tx1;
     ty1 = (k * (tz2-tz1) >> 15) + tz1;
     tz1 = ty1; continue;
   }
   if (c & 16){
     k   = ((tz2+tx2) << 15 )/(tx2-tx1-tz1+tz2);
     tx2 = (k * (tz2-tz1) >> 15) - tz2;
     ty2 = (k * (ty1-ty2) >> 15) + ty2;
     tz2 = -tx2; continue;
   }
   if (c & 32){
     k   = ((tz2-tx2) << 15 )/(tx1-tx2-tz1+tz2);
     tx2 = (k * (tz1-tz2) >> 15) + tz2;
     ty2 = (k * (ty1-ty2) >> 15) + ty2;
     tz2 = tx2; continue;
   }
   if (c & 64){
     k   = ((tz2+ty2) << 15 )/(ty2-ty1-tz1+tz2);
     tx2 = (k * (tx1-tx2) >> 15) + tx2;
     ty2 = (k * (tz2-tz1) >> 15) - tz2;
     tz2 = -ty2; continue;
   }
   if (c & 128){
     k   = ((tz2-ty2) << 15 )/(ty1-ty2-tz1+tz2);
     tx2 = (k * (tx1-tx2) >> 15) + tx2;
     ty2 = (k * (tz1-tz2) >> 15) + tz2;
```

```
      tz2 = ty2; continue;
    }
  }
}
tastatur()
{
  while (nachricht = GetMsg(window3->UserPort)){
    ReplyMsg(nachricht);
    if (nachricht->Class == VANILLAKEY){
      code = nachricht->Code;
      if (code > 95) code -= 32;  /* Umwandlung in grosse Buchstaben */
      switch (code){
        case '1' : if (flugz < 202) geschw -= 2;    /* Bremse */
                   if (geschw < 0) geschw = 0; break;
        case '4' : qruder -=8; if (qruder <-128) qruder = -128; break;
        case '6' : qruder +=8; if (qruder > 128) qruder = 128; break;
        case '2' : hruder -=8; if (hruder <-128) hruder = -128; break;
        case '8' : hruder +=8; if (hruder > 128) hruder = 128; break;
        case '5' : qruder = sruder = 0; break;
        case '3' : gas -=4; if (gas < 0) gas = 0; break;
        case '9' : gas +=4; if (gas > 100) gas = 100; break;
        case '0' : sruder -=8; if (sruder <-128) sruder = -128; break;
        case '.' : sruder +=8; if (sruder > 128) sruder = 128; break;
        case 'T' : sbl = 0.000; cbl = 1.000; break;  /* Blickrichtungen */
        case 'Y' : sbl = 0.707; cbl = 0.707; break;
        case 'H' : sbl = 1.000; cbl = 0.000; break;
        case 'N' : sbl = 0.707; cbl =-0.707; break;
        case 'B' : sbl = 0.000; cbl =-1.000; break;
        case 'V' : sbl =-0.707; cbl =-0.707; break;
        case 'F' : sbl =-1.000; cbl = 0.000; break;
        case 'R' : sbl =-0.707; cbl = 0.707; break;
        case 27  : menue(); break;
      }  /* Ende von switch */
      if (det & 0x0100) sruder = qruder;
    }
  }
}
long eing(min,max)
long min,max;
{
  long neuwert;
  do{
    PF(" Neuer Wert (%ld...%ld) : ",min,max);
    gets(input); neuwert = atol(input);
  } while (neuwert < min || neuwert > max);
  return(neuwert);
}
menue()
{
  WBenchToFront();
  ActivateWindow(window1);
anzeigen:
  CLEAR;
  PF(" "); for (abst = 0;abst < 75; abst++) PF("=");
  PF("\n                                   F L U S I   1 . 0\n");
  PF(" "); for (abst = 0;abst < 75; abst++) PF("=");
  PF("\n\n");
  PF(" ( 1 )  Gitternetz. . . . . . . . an/aus : %d\n",(det & 0x01) == 0x01);
  PF(" ( 2 )  Startbahn . . . . . . . . an/aus : %d\n",(det & 0x02) == 0x02);
  PF(" ( 3 )  See+Insel . . . . . . . . an/aus : %d\n",(det & 0x04) == 0x04);
  PF(" ( 4 )  Stadt . . . . . . . . . . an/aus : %d\n",(det & 0x08) == 0x08);
  PF(" ( 5 )  Funktuerme. . . . . . . . an/aus : %d\n",(det & 0x10) == 0x10);
  PF(" ( 6 )  Windraeder. . . . . . . . an/aus : %d\n",(det & 0x20) == 0x20);
  PF(" ( 7 )  Wald. . . . . . . . . . . an/aus : %d\n",(det & 0x40) == 0x40);
  PF(" ( 8 )  Voegel. . . . . . . . . . an/aus : %d\n",(det & 0x80) == 0x80);
  PF(" ( K )  Kopplung Seiten+Querruder an/aus : %d\n",(det & 0x100)==0x100);
  PF(" ( X )  X-Koordinate in Meter (-100000...100000): %ld\n",flugx/100);
  PF(" ( Y )  Y-Koordinate in Meter (-100000...100000): %ld\n",flugy/100);
  PF(" ( Z )  Flughoehe in Meter        (  2...100000): %ld\n",flugz/100);
  PF(" ( S )  Steigungswinkel in Grad   ( -90......90): %d\n",(int)(st*57.3));
  PF(" ( A )  Azimutwinkel in Grad      (-180.....180): %d\n",(int)(az*57.3));
  PF(" ( H )  Haengewinkel in Grad      (-180.....180): %d\n",(int)(ha*57.3));
  PF(" ( G )  Geschwindigkeit in Knoten (   0.....200): %d\n",(int)geschw);
  PF(" ( L )  Landebahn-Laenge in Meter (1000..100000): %ld\n",bahnlaenge/50);
  PF(" ( B )  Landebahn-Breite in Meter (  10.....200): %ld\n",bahnbreite/50);
  PF(" ( U )  Startposition\n");
  PF(" ( I )  Landeanflug\n");
  PF(" ( W )  Weiterfliegen\n");
  PF(" ( Q )  Programm beenden\n\n");
  PF(" Ihre Eingabe :  ");
  gets(input); PF("\n");
  if (input[0] > 95) input[0] -= 32;  /* Wandlung in grosse Buchstaben */
  switch (input[0]){
    case '1' : det ^= 0x0001; break;
    case '2' : det ^= 0x0002; break;
    case '3' : det ^= 0x0004; break;
    case '4' : det ^= 0x0008; break;
    case '5' : det ^= 0x0010; break;
    case '6' : det ^= 0x0020; break;
    case '7' : det ^= 0x0040; break;
    case '8' : det ^= 0x0080; break;
    case 'K' : det ^= 0x0100; break;
    case 'X' : flugx = 100 * eing(-100000L,100000L); break;
    case 'Y' : flugy = 100 * eing(-100000L,100000L); break;
    case 'Z' : flugz = 100 * eing(2L,100000L); break;
    case 'S' : st = 0.01745 * eing(- 90L, 90L); break;
    case 'A' : az = 0.01745 * eing(-180L,180L); break;
    case 'H' : ha = 0.01745 * eing(-180L,180L); break;
    case 'G' : geschw = eing(0L,200L); break;
    case 'L' : bahnlaenge = 50 * eing(1000L,100000L); break;
    case 'B' : bahnbreite = 50 * eing(10L,200L); break;
    case 'U' : flugx = 0; flugy = -bahnlaenge; flugz = 200;
               st = 0; az = 0; ha = 0; sbl = 0; cbl = 1; gas = 0;
               geschw = 0; hruder = qruder = sruder = 0; break;
    case 'I' : flugx = -10000; flugy = -1000000; flugz = 85000;
               st = -0.09; az = 0.01; ha = -0.01; sbl = 0; cbl = 1;
               gas = 50; geschw = 80; hruder = qruder = sruder = 0; break;
    case 'Q' : ende = 1; CLEAR; PF("\n bye-bye !\n\n"); break;
  }
  if (input[0] != 'W' && input[0] != 'Q') goto anzeigen;
  CurrentTime(&altsek,&altmikrosek);
  WBenchToBack();
  ActivateWindow(window3);
}
crash()
{
  SetAPen(rp2,3L); RectFill(rp2,0L,0L,319L,159L);
  SetAPen(rp3,5L); RectFill(rp3,0L,0L,319L,159L);
  for (abst = 1; abst < 50; abst++){
    ScreenToFront(screen2);
    ScreenToFront(screen3);
  }
  hruder = sruder = qruder = 0; gas = 0; geschw = 0;
  sbl = 0; cbl = 1;
}
initinstrumente()
{
```

Listing. Ein toller Flugsimulator: »Flusi« (Fortsetzung)

```
xg2  SetAPen(rp4,0L);
fX   RectFill(rp4,0L,0L,319L,95L);       /* Hintergrund */
3n   SetAPen(rp4,1L);
eV   for (abst = 30; abst < 34; abst++)
Vr4    for (abst2 = 32; abst2 < 300; abst2 += 64)
UW6      if (abst2 != 160)
9m8        DrawEllipse(rp4,abst2,34L,abst,abst);
272  for (abst = 19; abst < 22; abst++)
W14    DrawEllipse(rp4,64L,75L,abst,abst);
I42  SetAPen(rp4,3L);
bx   for (abst2 = 32; abst2 < 300; abst2 += 64)
ac4    if (abst2 != 160)
eH6      AreaEllipse(rp4,abst2,34L,5L,5L);
K32  AreaEllipse(rp4,64L,75L,3L,3L);
r2   AreaEnd(rp4);
x9   ziff(31,12,0); ziff(48,32,5); ziff(52,32,0);
Oc   ziff(27,52,1); ziff(31,52,0); ziff(35,52,0);
x4   ziff(10,32,1); ziff(14,32,5); ziff(18,32,0);
nb   zeich( 94,12,0); zeich(114,32,1); zeich( 94,52,2); zeich
     ( 74,32,3);
Ju   ziff(223,12,0); ziff(235,15,1); ziff(243,24,2); ziff(243
     ,39,3);
NG   ziff(235,48,4); ziff(223,52,5); ziff(211,48,6); ziff(203
     ,39,7);
HL   ziff(203,24,8); ziff(211,15,9);
1z   ziff(266,32,0); ziff(272,18,5); ziff(272,45,5);
U3   ziff(284,12,1); ziff(288,12,0); ziff(284,52,1); ziff(288
     ,52,0);
Xb   ziff(297,18,1); ziff(301,18,5); ziff(297,45,1); ziff(301
     ,45,5);
pg   ziff(304,32,2); ziff(308,32,0);
SP   Move(rp4,288L,24L); Draw(rp4,288L,20L);
1p   Move(rp4,286L,22L); Draw(rp4,290L,22L);
gM   Move(rp4,286L,46L); Draw(rp4,290L,46L);
Lm   SetAPen(rp4,2L);                    /* rote kritische Ber
     eiche */
57   for (winkel = 5.60; winkel < 6.283; winkel += .01){
Sy4    Move(rp4,(long)(32+sin(winkel)*26),(long)(34-cos(winke
       l)*26));
rK     Draw(rp4,(long)(32+sin(winkel)*29),(long)(34-cos(winke
       l)*29));
Mr2  }
F1   for (winkel = 2; winkel < 2.8; winkel += .01){
Aq4    Move(rp4,(long)(64+sin(winkel)*17),(long)(75-cos(winke
       l)*17));
Q1     Draw(rp4,(long)(64+sin(winkel)*19),(long)(75-cos(winke
       l)*19));
Qv2  }
1X   SetAPen(rp4,3L);
JU   marken( 32,34,6, 4); marken( 32,34,3,20); marken( 32,34,
     1,40);
Tt   marken( 96,34,6, 8); marken( 96,34,2,16); marken( 96,34,
     1,32);
2a   marken(224,34,5,10); marken(224,34,2,50);
GA   marken(288,34,5, 8); marken(288,34,2,40);
tU   for (winkel = -2.8; winkel < 2.8; winkel += .3)
2Q4    WritePixel(rp4,(long)(65+sin(winkel)*18),(long)(76-cos
       (winkel)*18));
kU2  SetAPen(rp4,1L);
oV   RectFill(rp4,125L, 5L,195L,17L);
gF   RectFill(rp4,154L,17L,166L,56L);
ry   RectFill(rp4,125L,55L,195L,67L);
kT   SetAPen(rp4,0L);
mT   RectFill(rp4,127L, 7L,193L,15L);
Z7   RectFill(rp4,156L,19L,164L,53L);
pw   RectFill(rp4,127L,57L,193L,65L);
Om   SetAPen(rp4,3L);
zK   for (abst = 128; abst < 193; abst += 4)
H9   { WritePixel(rp4,abst, 7L); WritePixel(rp4,abst,15L);
F74    WritePixel(rp4,abst,57L); WritePixel(rp4,abst,65L); }
1g2  for (abst = 128; abst < 193; abst += 16)
JQ   { WritePixel(rp4,abst, 8L); WritePixel(rp4,abst,14L);
H94    WritePixel(rp4,abst,58L); WritePixel(rp4,abst,64L); }
JH2  for (abst = 20; abst < 53; abst += 4)
Z2   { WritePixel(rp4,156L,abst); WritePixel(rp4,164L,abst);
     }
tV   for (abst = 20; abst < 53; abst += 8)
e2   { WritePixel(rp4,157L,abst); WritePixel(rp4,163L,abst);
     }
Bx   SetAPen(rp4,3L);
bo   RectFill(rp4,200L,80L,290L,100L);
9u   SetAPen(rp4,2L);
Xr   Move(rp4,210L,90L);
Sp   SetDrMd(rp4,JAM1);
Nw     Text(rp4,"Flusi 1.0\0",9L);
xS0  }
J2   ziff(x,y,n)
Qs   int x,y,n;
wP   {
vY2    int zeile,spalte,hilf;
JU     hilf = ziffer[n];
w6     for (zeile = 0; zeile < 5; zeile++)
k64      for (spalte = 0; spalte < 3; spalte++)
oL6        if ((hilf <<= 1) & 0x8000)
Ye8          WritePixel(rp4,(long) x + spalte,(long) y + zeile)
             ;
7c0  }
21   zeich(x,y,n)
a2   int x,y,n;
6Z   {
Rc2    int  zeile,spalte;
F6     long hilf;
In     hilf = zeichen[n];
7H     for (zeile = 0; zeile < 5; zeile++)
1P4      for (spalte = 0; spalte < 5; spalte++)
B16        if ((hilf <<= 1) & 0x80000000)
Jp8          WritePixel(rp4,(long) x + spalte,(long) y + zeile)
             ;
In0  }
NK   marken(x,y,laenge,anzahl)
Es   int x,y,laenge,anzahl;
Hk   {
2E2    float winkel,siw,cow;
yV     laenge = 30 - laenge;
Bi     for (winkel = 0; winkel < zp; winkel += (zp/anzahl)){
1C4      siw = sin(winkel); cow = cos(winkel);
mV       Move(rp4,(long)(x + siw * laenge),(long)(y - cow * lae
         nge));
UE       Draw(rp4,(long)(x + siw * 29    ),(long)(y - cow * 29
           ));
Sx2    }
Ty0  }
D0   instrumente()
Ru   {
wV2    Move(rp4,zeiger0x,zeiger0y); Draw(rp4,64L,75L);        /*
        Drehzahlmesser */
Z5     winkel = upm * .0016 - 2.8;
w0     if (winkel > 2.8) winkel = 2.8;
mz     zeiger0x =  64 + (int)(sin(winkel) * 15);
WF     zeiger0y =  75 - (int)(cos(winkel) * 15);
YF     Move(rp4,zeiger0x,zeiger0y); Draw(rp4,64L,75L);
hL     Move(rp4,zeiger1x,zeiger1y); Draw(rp4,224L,34L);
       /* Hoehenmesser */
GC     Draw(rp4,zeiger2x,zeiger2y);
XP     winkel = flugz * .0001915;
Rs     zeiger1x = 224 + (int)(sin(winkel) * 23);
Es     zeiger1y =  34 - (int)(cos(winkel) * 23);
c7     winkel /= 10;
d6     zeiger2x = 224 + (int)(sin(winkel) * 15);
Q6     zeiger2y =  34 - (int)(cos(winkel) * 15);
tm     Move(rp4,zeiger1x,zeiger1y); Draw(rp4,224L,34L);
OK     Draw(rp4,zeiger2x,zeiger2y);
4j     RectFill(rp4,zeiger3x,zeiger3y,zeiger3x+2,zeiger3y+2);
1D     winkel /= 10;
1F     zeiger3x = 223 + (int)(sin(winkel) * 25);
YF     zeiger3y =  33 - (int)(cos(winkel) * 25);
8n     RectFill(rp4,zeiger3x,zeiger3y,zeiger3x+2,zeiger3y+2);
Pg     Move(rp4,zeiger4x,zeiger4y); Draw(rp4,32L,34L);     /* G
       eschwindigkeit */
DY     winkel = geschw * .031416;
q5     if (winkel > 6) winkel = 6; if (winkel < 0) winkel = 0
       ;
mw     zeiger4x =  32 + (int)(sin(winkel) * 22);
XD     zeiger4y =  34 - (int)(cos(winkel) * 22);
d8     Move(rp4,zeiger4x,zeiger4y); Draw(rp4,32L,34L);
fQ     Move(rp4,zeiger5x,zeiger5y); Draw(rp4,288L,34L);
         /* Variometer */
nk     winkel = vario * .1571;
Fu     if (winkel > 3) winkel = 3; if (winkel <-3) winkel =-3
       ;
7E     zeiger5x = 288 - (int)(cos(winkel) * 22);
EU     zeiger5y =  34 - (int)(sin(winkel) * 22);
Pm     Move(rp4,zeiger5x,zeiger5y); Draw(rp4,288L,34L);
gt     Move(rp4,zeiger6x,zeiger6y); Draw(rp4,96L,34L);
         /* Kompass */
Oy     zeiger6x =  96 + (int)(saz * 22);
Jw     zeiger6y =  34 - (int)(caz * 22);
vA     Move(rp4,zeiger6x,zeiger6y); Draw(rp4,96L,34L);
```

```
    RectFill(rp4,zeiger7x,10L,zeiger7x+2,12L);
      /* Querruder */
    zeiger7x = 159 + (qruder >> 2);
    RectFill(rp4,zeiger7x,10L,zeiger7x+2,12L);
    RectFill(rp4,159L,zeiger8y,161L,zeiger8y+2);
    /* Hoehenruder */
    zeiger8y =  35 + (hruder >> 3);
    RectFill(rp4,159L,zeiger8y,161L,zeiger8y+2);
    RectFill(rp4,zeiger9x,60L,zeiger9x+2,62L);
    /* Seitenruder */
    zeiger9x = 159 + (sruder >> 2);
    RectFill(rp4,zeiger9x,60L,zeiger9x+2,62L);
  }
  farben(zeiger)
  BYTE *zeiger;
  {
    SetRGB4(zeiger,0L, 0L, 0L, 0L);  /* schwarz   starke Lin
    ien,Rand  */
    SetRGB4(zeiger,1L, 0L,12L, 0L);  /* gruen     Land
            */
    SetRGB4(zeiger,2L, 9L, 9L, 9L);  /* grau      Landebahn
            */
    SetRGB4(zeiger,3L,15L, 0L, 0L);  /* rot       Windraeder
    ,Funkturm */
    SetRGB4(zeiger,4L, 0L,10L, 0L);  /* gruen     schwache L
    inien     */
    SetRGB4(zeiger,5L,15L,15L,15L);  /* weiss     Windraeder
    ,Tuerme   */
    SetRGB4(zeiger,6L, 0L, 0L,10L);  /* blau      Wasser
            */
    SetRGB4(zeiger,7L, 0L, 9L,15L);  /* hellblau  Himmel
            */
  }
  main()
  {
    GfxBase = (struct GfxBase*) OpenLibrary("graphics.librar
    y",0L);
    if (GfxBase == NULL) goto cleanup;
    IntuitionBase = (struct IntuitionBase*) OpenLibrary("int
    uition.library",0L);
    if (IntuitionBase == NULL) goto cleanup;
    window1 = IntuitionBase->ActiveWindow;
    screen2 = OpenScreen(&newscreen);
    if (screen2 == NULL) goto cleanup;
    newwindow.Screen = screen2;
    window2 = OpenWindow(&newwindow);
    if (window2 == NULL) goto cleanup;
    rp2 = window2->RPort;
    farben(&screen2->ViewPort);
    screen3 = OpenScreen(&newscreen);
    if (screen3 == NULL) goto cleanup;
    newwindow.Screen = screen3;
    window3 = OpenWindow(&newwindow);
    if (window3 == NULL) goto cleanup;
    rp3 = window3->RPort;
    farben(&screen3->ViewPort);
    screen4 = OpenScreen(&newscreen4);
    if (screen4 == NULL) goto cleanup;
    newwindow4.Screen = screen4;
    window4 = OpenWindow(&newwindow4);
    if (window4 == NULL) goto cleanup;
    rp4 = window4->RPort;
    SetRGB4(&screen4->ViewPort,0L, 7L, 7L, 7L); /* grau
    Hintergrund       */
    SetRGB4(&screen4->ViewPort,1L, 0L, 0L, 0L); /* schwarz
    Instrumente       */
    SetRGB4(&screen4->ViewPort,2L,15L, 0L, 0L); /* rot
    kritische Bereiche */
    SetRGB4(&screen4->ViewPort,3L,15L,13L, 0L); /* gelb
    Zeiger,Ziffern     */
    InitArea(&a_info,&areabuf[0],100L);
    area_plane = AllocRaster(320L,256L);
    if (area_plane == NULL) goto cleanup;
    rp2->AreaInfo = &a_info; rp2->TmpRas = &t_ras;
    rp3->AreaInfo = &a_info; rp3->TmpRas = &t_ras;
    rp4->AreaInfo = &a_info; rp4->TmpRas = &t_ras;
    InitTmpRas(&t_ras,area_plane,(long)RASSIZE(320,256));
    maus = AllocMem(8L,0x00010003);        /* Chip-Ram */
    SetPointer(window1,maus,0L,0L,0L,0L);  /* Mauszeiger weg
     */
    SetPointer(window2,maus,0L,0L,0L,0L);
    SetPointer(window3,maus,0L,0L,0L,0L);
    SetPointer(window4,maus,0L,0L,0L,0L);
    ph=2*atan(1.0);      /* PI/2 */
    pi=4*atan(1.0);      /* PI   */
    zp=8*atan(1.0);      /* 2*PI */
    initinstrumente();
    SetDrMd(rp4,COMPLEMENT|JAM1);
    CurrentTime(&altsek,&altmikrosek);
  startsituation:
    flugx =        0;     /* Koordinaten des Flugzeugs in cm
          */
    flugy = -700000;
    flugz =   30000;
    st = 0;               /* Steigungswinkel         (-PI/2..
    .PI/2) */
    ha = 0;               /* Haengewinkel              (-PI
    ...PI) */
    az = 0;               /* Flugrichtungswinkel       (-PI
    ...PI) */
    sbl = 0; cbl = 1;     /* sin() bzw. cos(Kopfwinkel des Pi
    loten) */
    bahnlaenge = 500000; /* halbe Laenge der Startbahn in cm
          */
    bahnbreite = 2500;   /* halbe Breite der Startbahn in cm
          */
    hruder = 0;           /* Stellung des Hoehenruders (-128.
    ..128) */
    qruder = 0;           /* Stellung des Querruders   (-128.
    ..128) */
    sruder = 0;           /* Stellung des Seitenruders (-128.
    ..128) */
    gas = 80;             /* Gas in Prozent            ( 0.
    ..100) */
    geschw = 85;          /* Geschwindigkeit in Knoten  ( 0.
    ..200) */
    for(;;){              /* Hauptschleife */
      if (ende) goto cleanup2;
      count++;
      ruder();
      ActivateWindow(window3); tastatur();
      CurrentTime(&sek,&mikrosek);
      zeitdiff = 1000000 * (sek - altsek) + (mikrosek - altm
      ikrosek);
      altsek = sek; altmikrosek = mikrosek;
      flugmechanik();
      if (zeitdiff != 0)            /* Steiggeschw. in 10
      0ft/min */
        vario = (19685 * (flugz - altflugz)) / (float)zeitdi
        ff;
      altflugz = flugz;
      if (fenster == 2){
        fenster=3; rp=rp3;
        landschaft(); instrumente();   /* neues Bild verdeck
        t zeichnen */
        WBenchToBack();
        ScreenToBack(screen2);          /* und dann darstelle
        n */
      }
      else{
        fenster=2; rp=rp2;
        landschaft(); instrumente();
        WBenchToBack();
        ScreenToBack(screen3);
      }
    }
  cleanup:
    PF("Fehler !");
  cleanup2:
    if (area_plane) FreeRaster(area_plane,320L,256L);
    ClearPointer(window1); ClearPointer(window2);
    ClearPointer(window3); ClearPointer(window4);
    FreeMem(maus,8L);
    if (window2) CloseWindow (window2);
    if (screen2) CloseScreen (screen2);
    if (window3) CloseWindow (window3);
    if (screen3) CloseScreen (screen3);
    if (window4) CloseWindow (window4);
    if (screen4) CloseScreen (screen4);
    if (IntuitionBase) CloseLibrary (IntuitionBase);
    if (GfxBase) CloseLibrary (GfxBase);
    exit(0);
  }
(C) 1989 M&T
```

Listing. Ein toller Flugsimulator mit allen Schikanen: »Flusi« (Schluß)

Ping-pong pur

»Ping Pong« (Listing 1) füllt eine Lücke im Spieleangebot für den Amiga. Es ist nämlich unseres Wissens nach die erste Tischtennis-Simulation für diesen Computer. Die Programmierung in Assembler verleiht dem Programm eine rasante Geschwindigkeit und bringt Schwung ins Spiel. Die dreidimensionale Grafik und der »klassische« Pingpong-Sound tragen ihren Teil dazu bei, daß das Spiel absolut realitätsnah ist und Sie an den Joystick fesseln wird (Bild 1).

»Ping Pong« ist außerdem eines jener wenigen Spiele, bei denen zwei Spieler gleichzeitig gegeneinander antreten können.

Action zu zweit

Alles, was Sie tun müssen, um den kleinen weißen Ball auch auf Ihrem Bildschirm zum Leben zu erwecken, ist, das Listing mit dem Checksummer einzugeben. Da das Programm in Assembler geschrieben wurde, wir aber möglichst vielen Lesern ermöglichen wollen, in den Genuß von »Ping Pong« zu kommen, haben wir uns entschlossen, das Listing als Basic-Lader abzudrucken. Es kann also unter Beachtung der Eingabehinweise (siehe Kasten) wie jedes andere Basic-Programm eingetippt werden. Nach dem ersten Start von »PingPong_Gen« wird das lauffähige Programm »Ping Pong« automatisch auf Diskette erzeugt (siehe Eingabehinweise im Kasten).

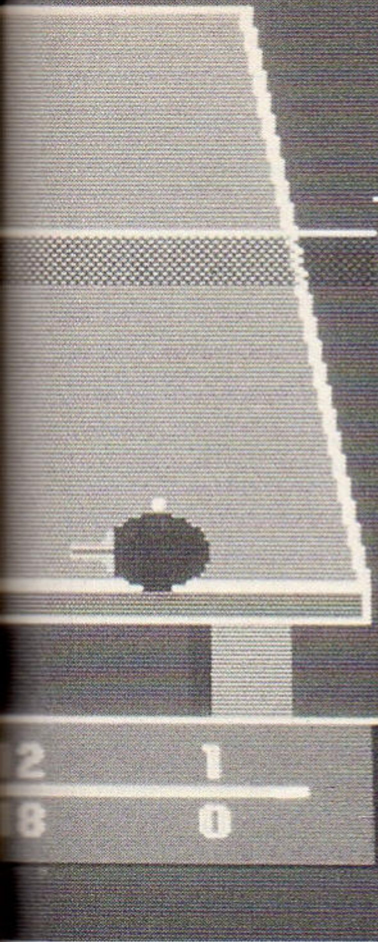
Das Spiel selbst läuft nach den üblichen Tischtennisregeln ab. Sieger ist der Spieler, der zwei Sätze gewinnt. Ein Satz ist gewonnen, wenn man 21 Punkte erreicht und mindestens zwei Punkte mehr auf seinem Konto als der Gegner hat.

Das Match beginnt mit der Angabe des Spielers im Vordergrund. Wenn er den Knopf an seinem Joystick drückt, fliegt der »simulierte Zelluloidball« in die Höhe, um sich dann gleich wieder dem Tischtennis-Schläger und dem Boden zu nähern. Der Joystick-Knopf muß nun ein weiteres Mal betätigt werden, um den Ball in die gegnerische Hälfte zu befördern — wichtig ist dabei der richtige Augenblick. Nach einigem Experimentie-

Bild 1. Sport, Spiel, Spannung mit der dreidimensionalen Tisch

Programmname:	PingPong_gen
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic
Bemerkung:	erzeugt lauffähiges Programm auf eingelegter Diskette (s. Kasten Seite 51)

```
Programmautor: S. Ören
----------------------
  1 OmO REM Generiert lauffähiges Programm
  2 Kq  REM >>PingPong<< auf der Diske
        tte
  3 cU  REM im Laufwerk df0:
  4 ci  CLS
  5 oJ  OPEN "df0:pingpong" FOR OUTPUT AS
        1
  6 DU  READ anz
  7 qc  FOR i=1 TO anz
  8 5p1  READ h$
  9 OD2   wert1=ASC(LEFT$(h$,1))
 10 dR    IF wert1>64 THEN wert1=wert1-87
           ELSE wert1=wert1-48
 11 HK    wert1=wert1*16
 12 9e    wert2=ASC(RIGHT$(h$,1))
 13 yr    IF wert2>64 THEN wert2=wert2-87
           ELSE wert2=wert2-48
 14 Rk    wert=wert1+wert2
 15 BI    PRINT #1,CHR$(wert);
 16 LQO NEXT
 17 5p  CLOSE 1
 18 2x  END
 19 0e  Werte:
 20 GE  DATA 9496
 21 rj  DATA 00,00,03,f3,00,00,00,00,00,00
 22 Se  DATA 00,01,00,00,00,00,00,00,00,00
 23 aR  DATA 00,00,07,e6,00,00,03,e9,00,00
 24 h2  DATA 07,e6,2c,78,00,04,20,3c,00,00
 25 zV  DATA 78,00,22,3c,00,03,00,02,4e,ae
 26 gS  DATA ff,3a,67,00,12,f6,23,c0,00,00
 27 r0  DATA 1f,30,20,3c,00,00,28,00,22,3c
 28 Bn  DATA 00,03,00,02,4e,ae,ff,3a,67,00
 29 6m  DATA 12,dc,23,c0,00,00,1f,3c,2c,7c
 30 or  DATA 00,00,1a,04,20,3c,00,00,18,c8
 31 eY  DATA 22,3c,00,00,19,10,24,3c,00,00
 32 q3  DATA 19,58,26,3c,00,00,19,64,28,3c
 33 gM  DATA 00,00,19,fc,2c,3c,00,00,19,70
 34 6x  DATA 2e,3c,00,00,19,b8,3d,40,00,06
 35 CO  DATA 48,40,3d,40,00,02,3d,41,00,0e
 36 xk  DATA 48,41,3d,41,00,0a,3d,42,00,16
 37 iU  DATA 48,42,3d,42,00,12,3d,43,00,1e
 38 TE  DATA 48,43,3d,43,00,1a,3d,44,00,26
 39 6z  DATA 3d,44,00,2e,48,44,3d,44,00,22
 40 TJ  DATA 3d,44,00,2a,3d,46,00,36,48,46
 41 Q6  DATA 3d,46,00,32,3d,47,00,3e,48,47
 42 53  DATA 3d,47,00,3a,20,3a,1e,78,22,00
 43 3i  DATA 24,01,06,81,00,00,28,00,06,82
 44 NY  DATA 00,00,50,00,23,c1,00,00,1f,34
 45 iF  DATA 23,c2,00,00,1f,38,26,3a,1e,64
 46 Dm  DATA 48,40,3d,40,00,42,48,40,3d,40
 47 hM  DATA 00,46,48,41,3d,41,00,4a,48,41
 48 73  DATA 3d,41,00,4e,48,42,3d,42,00,52
 49 Bp  DATA 48,42,3d,42,00,56,48,43,3d,43
 50 iJ  DATA 00,5a,48,43,3d,43,00,5e,3c,3c
 51 Qr  DATA 00,40,06,80,00,00,06,e8,20,40
 52 4x  DATA 20,3c,00,00,00,04,22,3c,00,00
 53 xV  DATA 00,09,61,00,16,1e,91,fc,00,00
 54 vY  DATA 00,02,20,3c,00,00,00,04,22,3c
 55 RX  DATA 00,00,00,0f,61,00,16,08,22,48
 56 HT  DATA e2,5e,20,3c,00,00,00,06,61,00
 57 o0  DATA 16,2e,20,49,e2,5e,d1,fc,00,00
 58 4m  DATA 50,00,20,3c,00,00,00,05,61,00
 59 hE  DATA 16,1a,22,48,d1,fc,00,00,00,2a
 60 xH  DATA 20,3c,00,00,00,14,61,00,16,08
 61 G9  DATA ea,5e,20,49,d1,fc,00,00,00,2c
 62 7E  DATA 20,3c,00,00,00,14,61,00,15,f4
 63 1G  DATA 20,49,91,fc,00,00,27,d4,20,3c
 64 6d  DATA 00,00,00,14,3c,3c,22,00,61,00
 65 wc  DATA 15,de,20,7a,1d,94,d1,fc,00,00
 66 4f  DATA 06,fa,3c,3c,08,00,20,3c,00,00
 67 Tb  DATA 00,04,22,3c,00,00,00,0b,61,00
 68 zi  DATA 15,a6,d1,fc,00,00,00,02,20,3c
 69 IO  DATA 00,00,00,04,22,3c,00,00,00,0d
 70 9d  DATA 61,00,15,90,22,48,e3,5e,20,3c
 71 HU  DATA 00,00,00,06,61,00,15,9c,20,49
 72 EL  DATA d1,fc,00,00,50,00,e3,5e,20,3c
 73 dj  DATA 00,00,00,05,61,00,15,88,3a,06
 74 Mm  DATA eb,5e,20,3c,00,00,00,14,22,48
 75 wv  DATA d1,fc,00,00,00,24,61,00,15,72
 76 OG  DATA 20,49,91,fc,00,00,27,dc,20,3c
 77 6g  DATA 00,00,00,14,3c,3c,01,10,61,00
 78 MA  DATA 15,5c,3c,05,20,49,d1,fc,00,00
 79 yF  DATA 00,26,20,3c,00,00,00,14,61,00
 80 3r  DATA 15,48,20,7a,1d,02,3c,3c,00,10
 81 VB  DATA d1,fc,00,00,06,e8,20,3c,00,00
 82 iq  DATA 00,04,22,3c,00,00,00,0b,61,00
 83 Vb  DATA 14,f6,91,fc,00,00,00,02,20,3c
 84 Xd  DATA 00,00,00,04,22,3c,00,00,00,0d
 85 hk  DATA 61,00,14,e0,91,fc,00,00,00,28
 86 6W  DATA 30,bc,00,00,91,fc,00,00,00,28
 87 St  DATA 30,bc,00,00,20,7a,1c,ba,d1,fc
 88 9K  DATA 00,00,06,fa,3c,3c,20,00,20,3c
 89 ci  DATA 00,00,00,04,22,3c,00,00,00,0d
 90 Os  DATA 61,00,14,c8,d1,fc,00,00,00,02
 91 ha  DATA 20,3c,00,00,00,04,22,3c,00,00
 92 VL  DATA 00,0b,61,00,14,b2,91,fc,00,00
 93 nz  DATA 00,28,30,bc,00,00,91,fc,00,00
 94 6M  DATA 00,28,30,bc,00,00,20,7a,1c,72
 95 R5  DATA d1,fc,00,00,07,18,3c,3c,00,22
 96 YO  DATA 20,3c,00,00,00,7e,61,00,14,a0
 97 ZC  DATA 20,7a,1c,5a,31,7c,00,00,06,e8
 98 yj  DATA 31,7c,00,00,06,fa,20,7a,1c,46
 99 sc  DATA 20,3c,00,00,77,fe,42,83,61,00
100 j5  DATA 15,9e,22,3c,00,00,00,0b,20,7a
101 8F  DATA 1c,30,d1,fc,00,00,0e,de,3c,3c
102 Eu  DATA ff,ff,20,3c,00,00,00,00,61,00
103 qa  DATA 14,70,91,fc,00,00,00,26,51,c9
104 20  DATA ff,ea,22,3c,00,00,00,0b,20,7a
105 tf  DATA 1c,0c,d1,fc,00,00,0e,de,3c,3c
106 Mu  DATA 00,00,20,3c,00,00,00,00,61,00
107 Rc  DATA 14,48,91,fc,00,00,00,26,51,c9
108 64  DATA ff,ea,22,3c,00,00,00,0b,20,7a
109 jg  DATA 1b,ec,d1,fc,00,00,0f,06,3c,3c
```



Erleben Sie Tischtennis hautnah auf dem Bildschirm. Tückische Angaben, Schmettern und Anschneiden des Balls sind in dieser Simulation realistisch umgesetzt. Bei zwei gleichwertigen Gegnern ergibt das packende Ballwechsel.

is-Simulation. Zwei Spieler treten dabei gegeneinander an.

ren findet man schnell den günstigsten Zeitpunkt heraus.

Schaffen Sie es, den Ball über das Netz zu befördern, ist der Gegner an der Reihe. Dieser muß ebenfalls den Ball mit seinem Schläger treffen und dabei gleichzeitig den Feuerknopf drücken. Mit der Bewegung des Joysticks während des Schlages nach rechts oder links kann der Abprallwinkel des Balls manipuliert werden. So wird das Anschneiden des Balls realisiert. Wird der Joystick nach vorne bewegt, steigert sich die Geschwindigkeit (Schmettern).

Nach jeweils fünf Punkten wechselt das Aufgaberecht. Ist ein Satz beendet, erfolgt ein Seitenwechsel. Spieler 1 führt den braunen Schläger und verwendet den Joystick in Port 1. Der zweite Spieler steuert seinen (roten) Schläger mit dem Joystick in Port 2.

Anfangs ist es sicherlich etwas schwierig, beispielsweise das Anschneiden des Balls geschickt einzusetzen. Nach kurzer Zeit werden Sie den Ball immer besser unter Kontrolle haben und Ball und Gegner beherrschen.

Martin Jobst/kn

Eingabehinweise

Starten Sie zur Generierung des lauffähigen Programms Amiga-Basic, das sich auf der, jedem Amiga beiliegenden, »Extras«-Diskette befindet. Geben Sie im Direktmodus die Zeile

```
CLEAR,35000
```

ein, gefolgt von <RETURN>. Laden Sie anschließend »Ping Pong_Gen« und entfernen den Schreibschutz auf der Diskette im eingebauten Laufwerk. Starten Sie nun die Programmausführung mit RUN. Jetzt leuchtet die LED des Laufwerks df0:, das Programm »Ping Pong« wird auf die eingelegte Diskette geschrieben. Nachdem sich Amiga-Basic mit »O.K.« meldet, verlassen Sie mit QUIT den Basic-Interpreter. Das auf diese Weise erzeugte Programm »Ping Pong« kann von jetzt an vom CLI aus gestartet werden. Wie man Programme vom CLI oder das CLI selbst aufruft, steht in Ihrem Amiga-Handbuch.

Besitzer einer Speichererweiterung oder eines Amiga 2000 sollten beachten, daß »Ping Pong« nur ohne Zusatzspeicher lauffähig ist. Dieser muß also vor dem Start unbedingt mit »NoFastMem« oder hardwaremäßig abgeschaltet werden.

```
110 s8  DATA 55,55,20,3c,00,00,00,0a,61,00
111 zM  DATA 14,20,91,fc,00,00,01,b6,51,c9
112 HH  DATA ff,ea,20,3c,00,00,00,1e,20,7a
113 vB  DATA 1b,b8,d1,fc,00,00,1f,be,3c,3c
114 ga  DATA c0,00,61,00,13,f0,22,3c,00,00
115 Oh  DATA 00,0b,20,7a,1b,ac,d1,fc,00,00
116 qG  DATA 1f,be,3c,3c,ff,ff,20,3c,00,00
117 Cr  DATA 00,1e,61,00,13,e0,91,fc,00,00
118 70  DATA 04,d6,51,c9,ff,ea,22,3c,00,00
119 fU  DATA 00,0b,20,7a,1b,80,d1,fc,00,00
120 uK  DATA 1f,be,3c,3c,ff,ff,20,3c,00,00
121 YI  DATA 00,1e,61,00,13,b8,91,fc,00,00
122 z0  DATA 04,d6,51,c9,ff,ea,20,3c,00,00
123 07  DATA 00,1e,20,7a,1b,58,d1,fc,00,00
124 aW  DATA 1f,d4,3c,3c,ff,fc,61,00,13,88
125 zj  DATA 22,3c,00,00,00,0b,20,7a,1b,38
126 T3  DATA d1,fc,00,00,1f,96,3c,3c,ff,ff
127 AT  DATA 20,3c,00,00,00,00,61,00,13,78
128 w5  DATA 91,fc,00,00,00,26,51,c9,ff,ea
129 bF  DATA 22,3c,00,00,00,0b,20,7a,1b,14
130 ZM  DATA d1,fc,00,00,24,96,3c,3c,ff,ff
131 Sb  DATA 20,3c,00,00,00,00,61,00,13,50
132 09  DATA 91,fc,00,00,00,26,51,c9,ff,ea
133 nM  DATA 22,3c,00,00,00,09,20,7a,1a,ec
134 Yv  DATA d1,fc,00,00,21,f0,3c,3c,ff,ff
135 1G  DATA 20,3c,00,00,00,01,61,00,13,28
136 PL  DATA 91,fc,00,00,00,4e,51,c9,ff,ea
137 JX  DATA 20,7a,1a,ca,d1,fc,00,00,20,88
138 kj  DATA 20,3c,00,00,00,06,22,7c,00,00
139 Mj  DATA 1a,68,61,00,13,14,20,7a,1a,b0
140 SG  DATA d1,fc,00,00,22,90,20,3c,00,00
141 ef  DATA 00,06,22,7c,00,00,1a,68,61,00
142 fZ  DATA 12,fa,20,7a,1a,96,d1,fc,00,00
143 Ck  DATA 20,8e,20,3c,00,00,00,03,22,7c
144 Ic  DATA 00,00,1a,92,61,00,13,06,20,7a
145 NL  DATA 1a,7c,d1,fc,00,00,22,96,20,3c
146 ec  DATA 00,00,00,03,22,7c,00,00,1a,9a
147 S8  DATA 61,00,12,ec,13,fc,00,15,00,00
148 MA  DATA 1f,90,61,00,0e,80,33,fc,01,2c
149 eU  DATA 00,00,1f,58,13,fc,00,44,00,00
150 6K  DATA 1f,81,33,fc,00,b4,00,00,1f,5c
151 fy  DATA 13,fc,00,c6,00,00,1f,80,2c,78
152 VM  DATA 00,04,4e,ae,ff,7c,4b,f9,00,df
153 fe  DATA f0,00,3b,7c,03,ef,00,96,2b,7c
154 CL  DATA 00,00,1a,f2,00,a0,2b,7c,00,00
155 Xc  DATA 1a,e2,00,b0,2b,7c,02,1f,02,5f
156 5e  DATA 00,a4,3b,7c,00,08,00,b4,3b,7c
157 Vi  DATA 00,40,00,a8,2b,7c,01,90,00,40
158 Rh  DATA 00,b6,3b,7c,00,ff,00,9e,2b,7c
159 Us  DATA 00,00,1a,04,00,80,42,6d,00,88
160 vC  DATA 2b,7c,29,81,29,f2,00,8e,2b,7c
161 7F  DATA 00,38,00,d0,00,92,2b,7c,42,00
162 Sj  DATA 00,00,01,00,3b,7c,00,17,01,04
163 Ng  DATA 2b,7c,00,06,0f,ff,01,80,2b,7c
164 vt  DATA 00,00,00,90,01,84,2b,7c,07,77
165 jO  DATA 00,60,01,88,2b,7c,03,33,00,00
166 1C  DATA 01,8c,2b,7c,0f,a6,0f,00,01,a2
167 UE  DATA 3b,7c,0b,40,01,a6,2b,7c,0f,ff
168 8J  DATA 00,00,01,aa,2b,7c,0f,a6,06,00
169 6W  DATA 01,ba,3b,7c,0b,40,01,be,20,7c
170 yu  DATA 00,df,f1,90,20,3c,00,00,00,07
171 d2  DATA 30,bc,00,00,d1,fc,00,00,00,02
172 FU  DATA 51,c8,ff,f4,2b,7c,00,0f,0f,ff
173 Qk  DATA 01,98,3b,7c,0f,f0,01,9c,3b,7c
174 28  DATA 83,a0,00,96,22,3a,19,78,10,3a
175 dd  DATA 19,9d,0c,39,00,01,00,00,1f,93
176 Jx  DATA 67,00,00,0e,34,39,00,df,f0,0a
177 Wq  DATA 4e,f9,00,00,06,04,34,39,00,df
178 gD  DATA f0,0c,08,02,00,01,67,00,00,1e
179 o1  DATA 48,41,0c,41,01,90,62,00,00,12
180 Dx  DATA 48,41,06,81,00,00,55,56,23,c1
181 34  DATA 00,00,1f,58,48,41,48,41,08,02
182 Um  DATA 00,09,67,00,00,1e,48,41,0c,41
183 Um  DATA 00,82,63,00,00,12,48,41,04,81
184 5D  DATA 00,00,55,56,23,c1,00,00,1f,58
185 8P  DATA 48,41,48,41,22,7c,00,00,1f,40
186 uF  DATA 24,7c,00,00,1f,44,12,80,14,80
187 3N  DATA 48,41,08,01,00,00,67,00,00,14
188 A1  DATA 13,7c,00,01,00,03,15,7c,00,01
189 Sp  DATA 00,03,4e,f9,00,00,06,84,13,7c
190 Zx  DATA 00,00,00,03,15,7c,00,00,00,03
191 5J  DATA e2,49,13,41,00,01,06,01,00,08
192 sQ  DATA 15,41,00,01,06,00,00,11,13,40
193 XX  DATA 00,02,04,00,00,01,15,40,00,02
194 vW  DATA 22,3a,18,b8,10,3a,18,d8,0c,39
195 XJ  DATA 00,01,00,00,1f,93,67,00,00,0e
196 we  DATA 34,39,00,df,f0,0c,4e,f9,00,00
197 Zp  DATA 06,c8,34,39,00,df,f0,0a,08,02
198 ym  DATA 00,01,67,00,00,1e,48,41,0c,41
199 g9  DATA 01,90,62,00,00,12,48,41,06,81
200 od  DATA 00,00,55,56,23,c1,00,00,1f,5c
201 Rs  DATA 48,41,48,41,08,02,00,09,67,00
202 5c  DATA 00,1e,48,41,0c,41,00,82,63,00
203 10  DATA 00,12,48,41,04,81,00,00,55,56
204 6Z  DATA 23,c1,00,00,1f,5c,48,41,48,41
205 jY  DATA 22,7c,00,00,1f,50,24,7c,00,00
206 3X  DATA 1f,54,12,80,14,80,48,41,08,01
207 01  DATA 00,00,67,00,00,14,13,7c,00,01
208 t6  DATA 00,03,15,7c,00,01,00,03,4e,f9
209 10  DATA 00,00,07,48,13,7c,00,00,00,03
210 8N  DATA 15,7c,00,00,00,03,e2,59,13,41
211 gG  DATA 00,01,50,01,15,41,00,01,06,00
212 F5  DATA 00,11,13,40,00,02,15,40,00,02
213 kk  DATA 0c,39,00,01,00,00,1f,83,67,00
214 cs  DATA 00,c6,0c,39,00,01,00,00,1f,8d
215 1L  DATA 66,00,00,32,0c,39,00,01,00,00
216 BM  DATA 1f,93,67,00,00,14,08,39,00,07
217 Lp  DATA 00,bf,e0,01,66,00,00,68,4e,f9
218 DM  DATA 00,00,07,d2,08,39,00,06,00,bf
219 7C  DATA e0,01,66,00,00,56,4e,f9,00,00
220 rj  DATA 07,d2,0c,39,00,01,00,00,1f,93
221 T2  DATA 67,00,00,14,08,39,00,06,00,bf
222 CH  DATA e0,01,66,00,00,38,4e,f9,00,00
223 sQ  DATA 07,d2,08,39,00,07,00,bf,e0,01
224 IU  DATA 66,00,00,26,23,fc,00,00,14,00
225 eD  DATA 00,00,1f,74,23,fc,00,20,00,00
226 hI  DATA 00,00,1f,78,13,fc,00,01,00,00
227 v0  DATA 1f,83,13,fc,00,01,00,00,1f,82
228 7t  DATA 0c,39,00,01,00,00,1f,8d,66,00
229 UK  DATA 00,1c,36,3a,17,58,06,43,00,12
230 Zu  DATA 61,00,0f,1e,33,fc,d5,00,00,00
```

Listing 1. Geben Sie »PingPong_Gen« bitte mit dem Checksummer (Seite 143) ein

```
231 on  DATA 1f,72,4e,f9,00,00,08,30,36,3a
232 3K  DATA 17,3a,06,43,00,09,61,00,0f,04
233 yS  DATA 33,fc,55,00,00,00,1f,72,20,3a
234 ng  DATA 17,42,91,b9,00,00,1f,78,20,3a
235 QS  DATA 17,3c,d1,b9,00,00,1f,7c,30,3a
236 DK  DATA 17,1e,d1,79,00,00,1f,72,18,3a
237 uV  DATA 17,22,20,3a,17,0c,d1,b9,00,00
238 pI  DATA 1f,66,20,3a,17,08,d1,b9,00,00
239 qh  DATA 1f,6e,36,3a,17,06,0c,39,00,46
240 fF  DATA 00,00,1f,72,62,00,00,12,13,fc
241 J9  DATA 00,01,00,00,1f,86,13,fc,00,01
242 GN  DATA 00,00,1f,8b,0c,39,00,eb,00,00
243 QJ  DATA 1f,72,65,00,00,12,13,fc,00,01
244 5b  DATA 00,00,1f,86,13,fc,00,02,00,00
245 vM  DATA 1f,8b,0c,39,00,3c,00,00,1f,7c
246 1w  DATA 63,00,01,1a,0c,39,00,01,00,00
247 yC  DATA 1f,82,66,00,00,2a,0c,39,00,55
248 6U  DATA 00,00,1f,72,66,00,00,0a,13,fc
249 Vd  DATA 00,01,00,00,1f,86,0c,39,00,d5
250 8W  DATA 00,00,1f,72,66,00,00,0a,13,fc
251 U8  DATA 00,01,00,00,1f,86,23,fc,00,1a
252 SN  DATA 00,00,00,00,1f,78,23,fc,00,00
253 5T  DATA 18,00,00,00,1f,74,0c,39,00,01
254 sz  DATA 00,00,1f,8c,66,00,00,0a,13,fc
255 DE  DATA 00,01,00,00,1f,86,0c,04,00,d6
256 JU  DATA 65,00,00,12,13,fc,00,01,00,00
257 cg  DATA 1f,86,13,fc,00,02,00,00,1f,8b
258 7W  DATA 0c,04,00,53,62,00,00,12,13,fc
259 bR  DATA 00,01,00,00,1f,86,13,fc,00,01
260 4i  DATA 00,00,1f,8b,3a,3c,00,00,3c,3c
261 IK  DATA 00,d6,9c,04,04,46,00,05,0c,46
262 QM  DATA 00,81,62,00,00,08,52,45,4e,fa
263 Z9  DATA ff,f0,3c,3c,00,b0,dc,45,bc,43
264 rr  DATA 65,00,00,0a,13,fc,00,01,00,00
265 Wa  DATA 1f,86,3c,3c,01,6e,9c,45,bc,43
266 kh  DATA 62,00,00,0a,13,fc,00,01,00,00
267 ta  DATA 1f,86,0c,39,00,00,00,00,1f,86
268 OH  DATA 66,00,00,3e,13,fc,00,01,00,00
269 pe  DATA 1f,8a,33,fc,01,90,00,df,f0,a6
270 hJ  DATA 33,fc,82,01,00,df,f0,96,13,fc
271 ch  DATA 00,14,00,00,1f,89,0c,39,00,00
272 gH  DATA 00,00,1f,82,67,00,00,12,13,fc
273 jJ  DATA 00,00,00,00,1f,8a,13,fc,00,00
274 ie  DATA 00,00,1f,82,04,39,00,01,00,00
275 Y6  DATA 1f,89,0c,39,00,00,00,00,1f,89
276 wh  DATA 66,00,00,0a,33,fc,00,01,00,df
277 H5  DATA f0,96,0c,39,00,93,00,00,1f,72
278 S7  DATA 67,00,00,14,0c,39,00,92,00,00
279 go  DATA 1f,72,67,00,00,08,4e,f9,00,00
280 eZ  DATA 0a,38,0c,39,00,0c,00,00,1f,7c
281 4M  DATA 62,00,00,2e,3a,3c,00,00,9a,79
282 fS  DATA 00,00,1f,64,33,c5,00,00,1f,64
283 sp  DATA 23,fc,00,00,30,00,00,00,1f,74
284 71  DATA 23,fc,00,10,00,00,00,00,1f,78
285 sx  DATA 13,fc,00,01,00,00,1f,8c,13,fc
286 nV  DATA 00,ff,00,00,1f,85,1a,04,9a,3a
287 14  DATA 15,38,13,fc,00,01,00,00,1f,84
288 8E  DATA 1c,05,9c,39,00,00,1f,80,20,3c
289 kW  DATA 00,00,00,04,1e,3c,00,00,bc,07
290 T4  DATA 66,00,00,08,13,c7,00,00,1f,85
291 Jg  DATA 52,07,51,c8,ff,f0,20,3c,00,00
292 Ap  DATA 00,04,1e,3c,00,0c,bc,07,66,00
293 ot  DATA 00,08,13,c0,00,00,1f,85,52,07
294 B4  DATA 51,c8,ff,f0,20,3c,00,00,00,04
295 Qe  DATA 1e,3c,00,06,bc,07,66,00,00,0a
296 Z8  DATA 13,fc,00,06,00,00,1f,85,52,07
297 bM  DATA 51,c8,ff,ee,20,3c,00,00,00,01
298 Pb  DATA 1e,3c,00,04,bc,07,66,00,00,0a
299 Y5  DATA 13,fc,00,04,00,00,1f,85,52,07
300 FF  DATA bc,07,66,00,00,0a,13,fc,00,04
301 5q  DATA 00,00,1f,85,5c,07,51,c8,ff,ec
302 ck  DATA 3c,3a,14,92,9c,7a,14,7c,0c,39
303 KU  DATA 00,00,00,00,1f,85,66,00,00,24
304 vP  DATA 0c,06,00,0e,65,00,00,1c,0c,06
305 P5  DATA 00,15,62,00,00,14,13,fc,00,00
306 xQ  DATA 00,00,1f,84,23,fc,00,00,00,00
307 Mw  DATA 00,00,1f,6a,0c,39,00,01,00,00
308 R6  DATA 1f,85,66,00,00,3e,0c,06,00,0b
309 WZ  DATA 65,00,00,36,0c,06,00,18,62,00
310 5M  DATA 00,2e,13,fc,00,00,00,00,1f,84
311 Vc  DATA 0c,06,00,0e,62,00,00,0c,23,fc
312 3n  DATA ff,ff,f9,00,00,00,1f,6a,0c,06
313 9Z  DATA 00,16,65,00,00,0c,23,fc,00,00
314 DI  DATA 07,00,00,00,1f,6a,0c,39,00,02
315 7q  DATA 00,00,1f,85,66,00,00,3e,0c,06
316 xL  DATA 00,0a,65,00,00,36,0c,06,00,19
317 ph  DATA 62,00,00,2e,13,fc,00,00,00,00
318 eE  DATA 1f,84,0c,06,00,0e,62,00,00,0c
319 6r  DATA 23,fc,ff,ff,f9,00,00,00,1f,6a
320 To  DATA 0c,06,00,16,65,00,00,0c,23,fc
321 x1  DATA 00,00,07,00,00,00,1f,6a,0c,39
322 nk  DATA 00,03,00,00,1f,85,66,00,00,50
323 TF  DATA 0c,06,00,0a,65,00,00,48,0c,06
324 yJ  DATA 00,1b,62,00,00,40,13,fc,00,00
325 9G  DATA 00,00,1f,84,0c,06,00,0e,62,00
326 mw  DATA 00,0c,23,fc,ff,ff,f9,00,00,00
327 2k  DATA 1f,6a,0c,06,00,16,65,00,00,0c
328 XG  DATA 23,fc,00,00,07,00,00,00,1f,6a
329 02  DATA 0c,06,00,1a,65,00,00,0c,23,fc
330 4c  DATA 00,00,0e,00,00,00,1f,6a,0c,39
331 2v  DATA 00,04,00,00,1f,85,66,00,00,50
332 c0  DATA 0c,06,00,0a,65,00,00,48,0c,06
333 Du  DATA 00,1c,62,00,00,40,13,fc,00,00
334 IP  DATA 00,00,1f,84,0c,06,00,0e,62,00
335 v5  DATA 00,0c,23,fc,ff,ff,f9,00,00,00
336 Bt  DATA 1f,6a,0c,06,00,16,65,00,00,0c
337 gP  DATA 23,fc,00,00,07,00,00,00,1f,6a
338 9B  DATA 0c,06,00,1a,65,00,00,0c,23,fc
339 D1  DATA 00,00,0e,00,00,00,1f,6a,0c,39
340 N8  DATA 00,06,00,00,1f,85,66,00,00,50
341 1X  DATA 0c,06,00,0a,65,00,00,48,0c,06
342 M3  DATA 00,1c,62,00,00,40,13,fc,00,00
343 RY  DATA 00,00,1f,84,0c,06,00,0e,62,00
344 4E  DATA 00,0c,23,fc,ff,ff,f9,00,00,00
345 K2  DATA 1f,6a,0c,06,00,16,65,00,00,0c
346 pY  DATA 23,fc,00,00,07,00,00,00,1f,6a
347 IK  DATA 0c,06,00,1a,65,00,00,0c,23,fc
348 Mu  DATA 00,00,0e,00,00,00,1f,6a,0c,39
349 xZ  DATA 00,d5,00,00,1f,72,67,00,00,46
350 yd  DATA 0c,39,00,d4,00,00,1f,72,67,00
351 wk  DATA 00,3a,0c,39,00,d6,00,00,1f,72
352 TZ  DATA 67,00,00,2e,0c,39,00,d3,00,00
353 BM  DATA 1f,72,67,00,00,22,0c,39,00,d2
354 w0  DATA 00,00,1f,72,67,00,00,16,0c,39
355 SZ  DATA 00,d7,00,00,1f,72,67,00,00,0a
356 1Q  DATA 13,fc,00,01,00,00,1f,84,0c,39
357 Ir  DATA 00,01,00,00,1f,84,67,00,00,cc
358 89  DATA 0c,39,00,01,00,00,1f,93,67,00
359 oI  DATA 00,14,08,39,00,07,00,bf,e0,01
360 AL  DATA 66,00,00,b4,4e,f9,00,00,0d,34
361 r4  DATA 08,39,00,06,00,bf,e0,01,66,00
362 mW  DATA 00,a2,33,fc,ff,dd,00,00,1f,64
363 q5  DATA 61,00,09,ae,0c,39,00,01,00,00
364 ul  DATA 1f,93,67,00,00,0e,34,39,00,df
365 jn  DATA f0,0c,4e,f9,00,00,0d,5e,34,39
366 8J  DATA 00,df,f0,0a,08,02,00,01,67,00
367 tt  DATA 00,0c,23,fc,ff,ff,ff,ff,00,00
368 MN  DATA 1f,60,08,02,00,09,67,00,00,0c
369 jZ  DATA 23,fc,00,00,00,01,00,00,1f,60
370 PB  DATA 0c,39,00,01,00,00,1f,8d,66,00
371 dF  DATA 00,1e,0c,39,00,01,00,00,1f,82
372 EQ  DATA 66,00,00,12,23,fc,00,00,00,00
373 Y3  DATA 00,00,1f,78,4e,f9,00,00,0d,d4
374 Rg  DATA 3c,02,e2,4e,b5,46,08,06,00,08
375 Lf  DATA 67,00,00,1e,33,fc,ff,c4,00,00
376 oF  DATA 1f,64,23,fc,00,20,00,00,00,00
377 Tg  DATA 1f,78,23,fc,00,00,30,00,00,00
378 Xx  DATA 1f,74,13,fc,00,01,00,00,1f,84
379 ia  DATA 13,fc,00,ff,00,00,1f,85,1c,05
380 Aq  DATA 9c,3a,11,99,20,3c,00,00,00,04
381 Jm  DATA 1e,3c,00,00,bc,07,66,00,00,08
382 Y3  DATA 13,c7,00,00,1f,85,52,07,51,c8
383 IY  DATA ff,f0,20,3c,00,00,00,04,1e,3c
384 wm  DATA 00,0c,bc,07,66,00,00,08,13,c0
385 Rb  DATA 00,00,1f,85,52,07,51,c8,ff,f0
386 nT  DATA 20,3c,00,00,00,04,1e,3c,00,06
387 oq  DATA bc,07,66,00,00,0a,13,fc,00,06
388 hh  DATA 00,00,1f,85,52,07,51,c8,ff,ee
389 X8  DATA 20,3c,00,00,00,01,1e,3c,00,04
390 hh  DATA bc,07,66,00,00,0a,13,fc,00,04
391 n4  DATA 00,00,1f,85,52,07,bc,07,66,00
392 8i  DATA 00,0a,13,fc,00,04,00,00,1f,85
393 Vp  DATA 5c,07,51,c8,ff,ec,3c,3a,10,fe
394 NE  DATA 9c,7a,10,e4,0c,39,00,00,00,00
395 y8  DATA 1f,85,66,00,00,24,0c,06,00,07
396 2e  DATA 65,00,00,1c,0c,06,00,0e,62,00
397 f5  DATA 00,14,13,fc,00,00,00,00,1f,84
398 61  DATA 23,fc,00,00,00,00,00,00,1f,6a
399 qr  DATA 0c,39,00,01,00,00,1f,85,66,00
400 9w  DATA 00,3e,0c,06,00,04,65,00,00,36
401 Vq  DATA 0c,06,00,11,62,00,00,2e,13,fc
402 FV  DATA 00,00,00,00,1f,84,0c,06,00,07
403 Dz  DATA 62,00,00,0c,23,fc,ff,ff,f9,00
404 mn  DATA 00,00,1f,6a,0c,06,00,0e,65,00
405 JF  DATA 00,0c,23,fc,00,00,07,00,00,00
406 93  DATA 1f,6a,0c,39,00,02,00,00,1f,85
407 0A  DATA 66,00,00,3e,0c,06,00,03,65,00
408 6q  DATA 00,36,0c,06,00,12,62,00,00,2e
409 9h  DATA 13,fc,00,00,00,00,1f,84,0c,06
410 L1  DATA 00,07,62,00,00,0c,23,fc,ff,ff
411 06  DATA f9,00,00,00,1f,6a,0c,06,00,0e
412 Xg  DATA 65,00,00,0c,23,fc,00,00,07,00
413 Co  DATA 00,00,1f,6a,0c,39,00,03,00,00
414 hk  DATA 1f,85,66,00,00,50,0c,06,00,01
415 0v  DATA 65,00,00,48,0c,06,00,12,62,00
416 pY  DATA 00,40,13,fc,00,00,00,00,1f,84
417 j6  DATA 0c,06,00,07,62,00,00,0c,23,fc
418 1V  DATA ff,ff,f9,00,00,00,1f,6a,0c,06
419 6m  DATA 00,03,62,00,00,0c,23,fc,ff,ff
420 Cn  DATA f2,00,00,00,1f,6a,0c,06,00,0e
421 gp  DATA 65,00,00,0c,23,fc,00,00,07,00
422 P2  DATA 00,00,1f,6a,0c,39,00,04,00,00
423 ln  DATA 1f,85,66,00,00,50,0c,06,00,00
424 94  DATA 65,00,00,48,0c,06,00,12,62,00
425 yh  DATA 00,40,13,fc,00,00,00,00,1f,84
426 sF  DATA 0c,06,00,07,62,00,00,0c,23,fc
427 S5  DATA ff,ff,f2,00,00,00,1f,6a,0c,06
428 Fv  DATA 00,03,62,00,00,0c,23,fc,ff,ff
429 Lw  DATA f2,00,00,00,1f,6a,0c,06,00,0e
430 py  DATA 65,00,00,0c,23,fc,00,00,07,00
431 gL  DATA 00,00,1f,6a,0c,39,00,06,00,00
432 uw  DATA 1f,85,66,00,00,50,0c,06,00,00
433 ID  DATA 65,00,00,48,0c,06,00,12,62,00
434 7q  DATA 00,40,13,fc,00,00,00,00,1f,84
435 1O  DATA 0c,06,00,07,62,00,00,0c,23,fc
436 3n  DATA ff,ff,f9,00,00,00,1f,6a,0c,06
437 04  DATA 00,03,62,00,00,0c,23,fc,ff,ff
438 U5  DATA f2,00,00,00,1f,6a,0c,06,00,0e
439 y7  DATA 65,00,00,0c,23,fc,00,00,07,00
440 0j  DATA 00,00,1f,6a,0c,39,00,55,00,00
441 zX  DATA 1f,72,67,00,00,46,0c,39,00,54
442 E1  DATA 00,00,1f,72,67,00,00,3a,0c,39
443 Rx  DATA 00,56,00,00,1f,72,67,00,00,2e
444 2m  DATA 0c,39,00,57,00,00,1f,72,67,00
445 PV  DATA 00,22,0c,39,00,58,00,00,1f,72
446 6k  DATA 67,00,00,16,0c,39,00,53,00,00
447 5k  DATA 1f,72,67,00,00,0a,13,fc,00,01
448 N6  DATA 00,00,1f,84,0c,39,00,01,00,00
449 NS  DATA 1f,84,67,00,00,d6,0c,39,00,01
450 CQ  DATA 00,00,1f,93,67,00,00,14,08,39
451 TE  DATA 00,06,00,bf,e0,01,66,00,00,be
452 Nb  DATA 4e,f9,00,00,10,c8,08,39,00,07
453 gU  DATA 00,bf,e0,01,66,00,00,ac,33,fc
454 NS  DATA 00,23,00,00,1f,64,61,00,06,1a
455 hi  DATA 0c,39,00,01,00,00,1f,93,67,00
456 aT  DATA 00,0e,34,39,00,df,f0,0a,4e,f9
457 6H  DATA 00,00,10,f2,34,39,00,df,f0,0c
458 83  DATA 08,02,00,01,67,00,00,0c,23,fc
459 Qj  DATA ff,ff,ff,ff,00,00,1f,60,08,02
460 r6  DATA 00,09,67,00,00,0c,23,fc,00,00
461 tz  DATA 00,01,00,00,1f,60,0c,39,00,01
462 ZQ  DATA 00,00,1f,8d,67,00,00,28,0c,39
463 aN  DATA 00,01,00,00,1f,82,66,00,00,1c
464 vt  DATA 23,fc,00,00,00,00,00,00,1f,78
465 ol  DATA 23,fc,00,00,30,00,00,00,1f,74
466 wu  DATA 4e,f9,00,00,11,72,3c,02,e2,4e
467 V4  DATA b5,46,08,06,00,00,67,00,00,1e
468 1d  DATA 33,fc,00,3c,00,00,1f,64,23,fc
469 mS  DATA 00,20,00,00,00,00,1f,78,23,fc
470 CE  DATA 00,00,30,00,00,00,1f,74,0c,39
471 PZ  DATA 00,01,00,00,1f,86,66,00,01,08
472 zk  DATA 0c,39,00,01,00,00,1f,8c,66,00
473 8x  DATA 00,26,0c,39,00,93,00,00,1f,72
474 fH  DATA 62,00,00,0e,52,39,00,00,1f,88
475 YO  DATA 4e,f9,00,00,12,74,52,39,00,00
476 Dh  DATA 1f,87,4e,f9,00,00,12,74,0c,39
477 G3  DATA 00,01,00,00,1f,8a,66,00,00,32
478 1l  DATA 0c,39,00,01,00,00,1f,8b,66,00
479 Kd  DATA 00,0e,52,39,00,00,1f,88,4e,f9
```

```
480 VT  DATA 00,00,12,74,0c,39,00,02,00,00
481 Sh  DATA 1f,8b,66,00,00,0e,52,39,00,00
482 Jn  DATA 1f,87,4e,f9,00,00,12,74,0c,39
483 QK  DATA 00,00,00,00,1f,8a,66,00,00,80
484 1z  DATA 0c,39,00,01,00,00,1f,82,66,00
485 K9  DATA 00,26,0c,39,00,93,00,00,1f,72
486 rT  DATA 62,00,00,0e,52,39,00,00,1f,88
487 ka  DATA 4e,f9,00,00,12,74,52,39,00,00
488 Pt  DATA 1f,87,4e,f9,00,00,12,74,0c,39
489 bf  DATA 00,93,00,00,1f,72,62,00,00,0e
490 wZ  DATA 52,39,00,00,1f,87,4e,f9,00,00
491 T6  DATA 12,74,52,39,00,00,1f,88,4e,f9
492 da  DATA 00,00,12,74,0c,39,00,01,00,00
493 et  DATA 1f,8b,66,00,00,0e,52,39,00,00
494 Vz  DATA 1f,87,4e,f9,00,00,12,74,0c,39
495 zm  DATA 00,02,00,00,1f,8b,66,00,00,08
496 lh  DATA 52,39,00,00,1f,88,61,00,00,e4
497 RT  DATA 0c,39,00,01,00,00,1f,94,67,00
498 s8  DATA 00,8a,22,7c,00,00,1f,48,24,7c
499 Dp  DATA 00,00,1f,4c,08,03,00,00,67,00
500 Ad  DATA 00,14,13,7c,00,01,00,03,15,7c
501 ru  DATA 00,01,00,03,4e,f9,00,00,12,b6
502 7w  DATA 13,7c,00,00,00,03,15,7c,00,00
503 pd  DATA 00,03,e2,4b,14,84,54,04,15,44
504 EB  DATA 00,02,15,43,00,01,12,85,54,05
505 pf  DATA 13,45,00,02,13,43,00,01,33,fc
506 fc  DATA a0,20,00,df,f0,9c,23,fa,0c,66
507 Ks  DATA 00,00,19,70,23,fa,0c,62,00,00
508 lu  DATA 19,b8,23,fa,0c,5e,00,00,19,58
509 6c  DATA 23,fa,0c,5a,00,00,19,64,23,fa
510 m5  DATA 0c,56,00,00,18,c8,23,fa,0c,52
511 18  DATA 00,00,19,10,4e,fa,f2,d4,2a,3c
512 qE  DATA 00,00,00,14,61,00,03,c2,51,cd
513 SX  DATA ff,fa,61,00,03,54,61,00,02,f8
514 bg  DATA 13,fc,00,00,00,00,1f,8d,13,fc
515 LA  DATA 00,00,00,00,1f,91,13,fc,00,00
516 Wm  DATA 00,00,1f,92,61,00,00,1e,13,fc
517 cW  DATA 00,00,00,00,1f,96,13,fc,00,00
518 Pn  DATA 00,00,1f,94,13,fc,00,00,00,00
519 pn  DATA 1f,95,4e,fa,f2,86,24,3c,00,00
520 VT  DATA 00,00,14,39,00,00,1f,87,d4,39
521 P6  DATA 00,00,1f,88,84,fc,00,05,48,42
522 Hc  DATA 0c,42,00,00,66,00,00,16,0c,39
523 bL  DATA 00,01,00,00,1f,95,67,00,00,0a
524 tK  DATA 08,79,00,00,00,00,1f,8d,0c,39
525 IM  DATA 00,01,00,00,1f,93,67,00,00,18
526 Tr  DATA 13,fa,0b,eb,00,00,1f,8e,13,fa
527 Wj  DATA 0b,e4,00,00,1f,8f,4e,f9,00,00
528 WZ  DATA 13,c0,13,fa,0b,d5,00,00,1f,8f
529 MI  DATA 13,fa,0b,ce,00,00,1f,8e,0c,39
530 oo  DATA 00,15,00,00,1f,87,65,00,00,18
531 EK  DATA 08,79,00,00,00,00,1f,8d,13,fc
532 HF  DATA 00,01,00,00,1f,95,4e,f9,00,00
533 P7  DATA 13,fe,0c,39,00,15,00,00,1f,88
534 WJ  DATA 65,00,00,12,08,79,00,00,00,00
535 wJ  DATA 1f,8d,13,fc,00,01,00,00,1f,95
536 dP  DATA 24,3c,00,00,00,00,14,3a,0b,88
537 s1  DATA b4,3a,0b,86,65,00,00,2e,94,39
538 mr  DATA 00,00,1f,8f,0c,02,00,01,66,00
539 p1  DATA 00,0e,52,39,00,00,1f,90,4e,f9
540 Gc  DATA 00,00,14,3c,08,79,00,00,00,00
541 1F  DATA 1f,93,52,39,00,00,1f,91,61,00
542 xb  DATA 02,36,14,3a,0b,51,b4,3a,0b,4e
543 Yz  DATA 65,00,00,30,94,39,00,00,1f,8e
544 E9  DATA 0c,02,00,01,66,00,00,0e,52,39
545 4m  DATA 00,00,1f,90,4e,f9,00,00,14,76
546 gc  DATA 08,79,00,00,00,00,1f,93,06,39
547 ww  DATA 00,01,00,00,1f,92,61,00,01,fc
548 43  DATA 0c,39,00,02,00,00,1f,91,66,00
549 V2  DATA 00,0a,13,fc,00,01,00,00,1f,94
550 AA  DATA 0c,39,00,02,00,00,1f,92,66,00
551 X4  DATA 00,0a,13,fc,00,01,00,00,1f,94
552 Jw  DATA 14,39,00,00,1f,91,d4,39,00,00
553 P3  DATA 1f,92,0c,02,00,02,66,00,00,2e
554 UY  DATA 0c,39,00,01,00,00,1f,96,67,00
555 Bh  DATA 00,22,0c,39,00,0a,00,00,1f,8e
556 i1  DATA 66,00,00,06,61,00,01,de,0c,39
557 ZM  DATA 00,0a,00,00,1f,8f,66,00,00,06
558 fe  DATA 61,00,01,ce,0c,39,00,01,00,00
559 1t  DATA 1f,93,67,00,00,18,33,fc,0f,00
560 wS  DATA 00,df,f1,a4,33,fc,06,00,00,df
561 X2  DATA f1,bc,4e,f9,00,00,15,10,33,fc
562 eg  DATA 06,00,00,df,f1,a4,33,fc,0f,00
563 OQ  DATA 00,df,f1,bc,14,3a,0a,7c,84,fc
564 fV  DATA 00,0a,20,7a,0a,16,d1,fc,00,00
565 wa  DATA 20,92,20,3c,00,00,00,03,61,00
566 JO  DATA 02,dc,61,00,02,b2,48,42,4e,f9
567 9o  DATA 00,00,15,3c,14,3a,0a,54,20,7a
568 xH  DATA 09,f2,d1,fc,00,00,20,94,20,3c
569 UL  DATA 00,00,00,03,61,00,02,b8,61,00
570 De  DATA 02,68,24,3c,00,00,00,00,14,3a
571 MJ  DATA 0a,33,84,fc,00,0a,20,7a,09,cc
572 nO  DATA d1,fc,00,00,22,9a,20,3c,00,00
573 7n  DATA 00,03,61,00,02,92,61,00,02,68
574 8r  DATA 48,42,4e,f9,00,00,15,86,14,3a
575 jS  DATA 0a,0b,20,7a,09,a8,d1,fc,00,00
576 xT  DATA 22,9c,20,3c,00,00,00,03,61,00
577 zG  DATA 02,6e,61,00,02,1e,20,7a,09,94
578 ke  DATA d1,fc,00,00,20,98,20,3c,00,00
579 wM  DATA 00,03,14,39,00,00,1f,91,61,00
580 cY  DATA 02,50,61,00,02,26,20,7a,09,76
581 rJ  DATA d1,fc,00,00,22,a0,20,3c,00,00
582 3U  DATA 00,03,14,39,00,00,1f,92,61,00
583 CD  DATA 02,32,61,00,02,08,23,fc,00,00
584 7H  DATA 00,00,00,00,19,58,23,fc,00,00
585 3j  DATA 00,00,00,00,19,64,33,fc,82,02
586 TA  DATA 00,df,f0,96,3b,7c,01,40,00,b6
587 Kh  DATA 61,00,00,d8,3b,7c,01,ab,00,b6
588 rK  DATA 61,00,00,ce,33,fc,00,02,00,df
589 0a  DATA f0,96,61,00,00,04,4e,75,13,fc
590 fQ  DATA 00,01,00,00,1f,82,13,fc,00,00
591 St  DATA 00,00,1f,83,13,fc,00,00,00,00
592 at  DATA 1f,86,23,fc,00,00,00,00,00,00
593 Kk  DATA 1f,74,23,fc,00,00,00,00,00,00
594 T7  DATA 1f,60,33,fc,00,00,00,00,1f,64
595 oJ  DATA 23,fc,00,00,00,00,00,00,1f,66
596 j1  DATA 13,fc,00,00,00,00,1f,8a,13,fc
597 0b  DATA 00,00,00,00,1f,8b,13,fc,00,00
598 x3  DATA 00,00,1f,8c,4e,75,13,fc,00,01
599 AO  DATA 00,00,1f,8d,13,fc,00,00,00,00
600 Ll  DATA 1f,95,13,fc,00,15,00,00,1f,90
601 6S  DATA 13,fc,00,00,00,00,1f,87,13,fc
602 3y  DATA 00,00,00,00,1f,88,13,fc,00,00
603 GV  DATA 00,00,1f,8e,13,fc,00,00,00,00
604 1k  DATA 1f,8f,4e,75,08,79,00,00,00,00
605 U3  DATA 1f,93,13,fc,00,01,00,00,1f,96
606 EG  DATA 14,3a,08,cb,13,fa,08,c8,00,00
607 eS  DATA 1f,87,13,c2,00,00,1f,88,08,79
608 kV  DATA 00,00,00,00,1f,8d,4e,75,24,3c
609 Pa  DATA 00,00,00,01,51,c8,ff,fe,20,3c
610 fO  DATA ff,ff,ff,ff,51,ca,ff,f4,4e,75
611 A4  DATA 13,fc,00,00,00,00,1f,8a,33,fc
612 5K  DATA 02,5f,00,df,f0,a6,33,fc,82,01
613 wZ  DATA 00,df,f0,96,13,fc,00,3c,00,00
614 dY  DATA 1f,89,23,fc,00,1a,00,00,00,00
615 va  DATA 1f,78,23,fc,00,00,18,00,00,00
616 VD  DATA 1f,74,23,fa,08,48,00,00,1f,66
617 yt  DATA 4e,75,33,c3,00,00,1f,6e,23,fc
618 a7  DATA 13,00,00,00,00,00,1f,7c,20,3c
619 ff  DATA 00,00,00,00,4e,75,30,86,d1,fc
620 yN  DATA 00,00,00,28,51,c8,ff,f6,20,3c
621 Qb  DATA 00,00,00,04,e3,5e,51,c9,ff,ea
622 Jc  DATA 4e,75,30,86,d1,fc,00,00,00,28
623 gf  DATA 51,c8,ff,f6,20,3c,00,00,00,04
624 GH  DATA e2,5e,51,c9,ff,ea,4e,75,30,86
625 y9  DATA d1,fc,00,00,00,28,51,c8,ff,f6
626 Ng  DATA 4e,75,30,86,d1,fc,00,00,00,28
627 Mb  DATA e3,5e,51,c8,ff,f4,4e,75,24,11
628 XT  DATA 20,82,d3,fc,00,00,00,04,d1,fc
629 MQ  DATA 00,00,00,04,34,11,30,82,d3,fc
630 KG  DATA 00,00,00,02,d1,fc,00,00,00,24
631 dG  DATA 51,c8,ff,de,4e,75,c4,38,00,0f
632 uR  DATA 14,11,10,82,d1,fc,00,00,00,28
633 Jd  DATA 14,29,00,01,10,82,d3,fc,00,00
634 w4  DATA 00,02,d1,fc,00,00,00,28,51,c8
635 QZ  DATA ff,de,4e,75,14,11,11,42,00,01
636 xl  DATA d1,fc,00,00,00,28,14,29,00,01
637 sj  DATA 11,42,00,01,d3,fc,00,00,00,02
638 qk  DATA d1,fc,00,00,00,28,51,c8,ff,de
639 7m  DATA 4e,75,0c,02,00,00,66,00,00,08
640 CX  DATA 22,7c,00,00,1a,da,0c,02,00,01
641 Uy  DATA 66,00,00,08,22,7c,00,00,1a,92
642 IR  DATA 0c,02,00,02,66,00,00,08,22,7c
643 D6  DATA 00,00,1a,9a,0c,02,00,03,66,00
644 qY  DATA 00,08,22,7c,00,00,1a,a2,0c,02
645 q0  DATA 00,04,66,00,00,08,22,7c,00,00
646 pI  DATA 1a,aa,0c,02,00,05,66,00,00,08
647 NO  DATA 22,7c,00,00,1a,b2,0c,02,00,06
648 2w  DATA 66,00,00,08,22,7c,00,00,1a,ba
649 Zn  DATA 0c,02,00,07,66,00,00,08,22,7c
650 Ch  DATA 00,00,1a,c2,0c,02,00,08,66,00
651 5a  DATA 00,08,22,7c,00,00,1a,ca,0c,02
652 RH  DATA 00,09,66,00,00,08,22,7c,00,00
653 CZ  DATA 1a,d2,4e,75,22,3c,00,00,00,07
654 Q6  DATA d1,fc,00,00,00,01,1c,10,08,06
655 R9  DATA 00,07,67,00,00,06,08,43,00,00
656 dB  DATA 08,03,00,00,67,00,00,06,08,c6
657 pr  DATA 00,07,e3,1e,51,c9,ff,e4,10,86
658 4g  DATA 51,c8,ff,d0,4e,75,a0,5a,b1,00
659 9I  DATA 00,00,00,01,00,00,00,0f,00,00
660 zs  DATA 00,1f,00,00,00,7f,00,80,00,7f
661 Np  DATA 00,80,00,7f,01,80,00,7f,ff,80
662 JP  DATA 00,7f,ff,80,ff,ff,ff,80,00,7f
663 PT  DATA 01,80,00,7f,00,80,00,7f,00,80
664 1W  DATA 00,7f,00,00,00,7f,00,00,00,1f
665 YP  DATA 00,00,00,0f,00,00,00,01,a0,5a
666 Um  DATA b1,00,00,00,fc,00,00,00,ff,80
667 zx  DATA 00,00,ff,c0,00,00,ff,f0,00,00
668 ny  DATA ff,f8,00,00,ff,f8,00,00,ff,fc
669 Ff  DATA 00,00,ff,fc,00,00,ff,fc,00,00
670 b0  DATA ff,fc,00,00,ff,fc,00,00,ff,f8
671 bV  DATA 00,00,ff,f8,00,00,ff,f0,00,00
672 HJ  DATA ff,c0,00,00,ff,80,00,00,fc,00
673 0j  DATA a0,6a,a2,00,c0,00,00,00,c0,00
674 yq  DATA 00,00,a8,6a,aa,00,00,00,c0,00
675 Bf  DATA 00,00,c0,00,a0,5a,b1,00,00,00
676 t4  DATA 00,fe,00,00,07,ff,00,00,0f,ff
677 PL  DATA 00,00,3f,ff,00,00,7f,ff,00,00
678 Mb  DATA 7f,ff,00,00,ff,ff,00,00,ff,ff
679 hD  DATA 00,00,ff,ff,00,00,ff,ff,00,00
680 Kb  DATA ff,ff,00,00,7f,ff,00,00,7f,ff
681 uj  DATA 00,00,3f,ff,00,00,0f,ff,00,00
682 Rc  DATA 07,ff,00,00,00,fe,a0,62,b0,00
683 8D  DATA 00,00,00,00,00,00,c0,00,00,00
684 kF  DATA e0,00,00,00,f8,00,04,00,f8,00
685 oO  DATA 04,00,f8,00,06,00,f8,00,07,fc
686 df  DATA f8,00,07,fc,ff,fc,07,fc,f8,00
687 5v  DATA 06,00,f8,00,04,00,f8,00,04,00
688 UK  DATA f8,00,00,00,f8,00,00,00,e0,00
689 af  DATA 00,00,c0,00,00,00,00,00,00,00
690 ir  DATA 00,00,01,20,00,00,01,22,00,00
691 zM  DATA 01,24,00,00,01,26,00,00,01,28
692 UD  DATA 00,00,01,2a,00,00,01,2c,00,00
693 58  DATA 01,2e,00,00,01,30,00,00,01,32
694 JX  DATA 00,00,01,34,00,00,01,36,00,00
695 Ai  DATA 01,38,00,00,01,3a,00,00,01,3c
696 re  DATA 00,00,01,3e,00,00,00,e0,00,00
697 WN  DATA 00,e2,00,00,00,e4,00,00,00,e6
698 Nz  DATA 00,00,00,e8,00,00,00,ea,00,00
699 ZE  DATA 00,ec,00,00,00,ee,00,00,ff,ff
700 uu  DATA ff,fe,7c,c1,e6,6f,be,00,66,c3
701 nW  DATA 36,6c,33,00,66,c3,33,cc,33,00
702 MI  DATA 6c,c3,f1,8f,36,00,60,c3,31,8c
703 g6  DATA 36,00,60,c3,31,8c,33,00,60,fb
704 Sh  DATA 31,8f,33,00,38,18,18,18,18,18
705 uN  DATA 18,00,3c,66,06,1c,30,60,7e,00
706 wa  DATA 3c,66,06,1c,06,66,3c,00,66,66
707 7k  DATA 66,3e,06,06,06,00,7e,60,7c,06
708 Bz  DATA 06,06,7c,00,3c,60,60,7c,66,66
709 6J  DATA 3c,00,7c,0c,0c,18,18,30,30,00
710 2f  DATA 3c,66,66,3c,66,66,3c,00,3c,66
711 ii  DATA 66,3e,06,06,3c,00,3c,66,66,66
712 hW  DATA 66,66,3c,00,00,31,5a,75,7f,75
713 JS  DATA 5a,31,00,cf,a6,8b,81,8b,a6,cf
714 zy  DATA 70,80,3d,b7,64,a0,80,80,80,9f
715 11  DATA 5c,d8,80,80,80,bd,3f,61,10,88
716 mN  DATA 80,80,9a,3f,7f,84,80,a4,f3,ff
717 0j  DATA 3b,42,d8,80,80,3f,74,24,c0,80
718 0s  DATA 80,97,4b,68,f0,84,80,1f,34,d0
719 J2  DATA e4,80,bf,bf,d7,00,2f,e8,80,80
720 Za  DATA cf,3f,20,a6,8b,80,82,ff,40,e8
721 7K  DATA 80,80,cf,20,00,f7,e0,a0,80,9f
722 Ue  DATA 30,02,cb,ba,80,9b,df,fd,c0,a9
723 4z  DATA 99,a6,d7,19,1c,a4,80,9f,ef,cc
724 dh  DATA f3,e0,b0,94,8b,d8,04,d0,bf,ac
725 XI  DATA a8,af,0f,08,ca,87,90,b7,ef,f2
```

Listing 1. (Fortsetzung)

```
726 Ps DATA fd,a0,80,a7,f2,f3,cc,a8,92,a7
727 I8 DATA 08,f1,c8,b7,8c,a2,f7,07,e0,d5
728 cN DATA e9,b8,93,b0,d6,d7,ca,df,b0,94
729 9V DATA d3,e1,d0,b3,94,c9,cf,02,fe,d0
730 6K DATA 80,b7,b0,f7,f1,c8,9e,97,ce,be
731 su DATA df,d0,a7,c5,d0,e5,cc,bc,d7,d0
732 29 DATA ec,80,b3,ff,b8,b7,e5,c2,88,9b
733 dL DATA 17,28,d8,80,97,bd,f7,f3,a8,90
734 1f DATA 99,c9,fa,f4,c8,97,9f,ad,db,eb
735 Gw DATA e0,d1,8f,c4,f5,f4,ba,ae,b5,d9
736 BH DATA a7,ca,e7,e9,d8,bf,a3,a6,ef,00
737 uU DATA e2,89,95,bf,db,dc,d6,b0,84,a7
738 PE DATA f1,f3,d2,b0,9b,c7,c0,df,00,b8
739 Zf DATA 9d,b0,bf,bf,e0,e5,ac,a2,bf,e7
740 NR DATA ec,d9,cd,b0,a7,c4,df,eb,d8,b3
741 TO DATA a7,d0,e7,d8,bc,af,b8,c7,c1,b8
742 MJ DATA cb,bc,a0,93,bf,e0,df,c8,b2,bd
743 cr DATA c0,d7,e9,d0,a4,b7,c8,cf,d2,ce
744 dx DATA c9,b6,bf,df,ea,d5,bf,c5,c2,d7
745 v7 DATA dc,cf,c9,cf,cf,c0,bb,dd,d8,c1
746 lc DATA b3,c0,ab,c7,cf,cd,b6,b3,b9,c8
747 W1 DATA c9,c0,b4,b3,c5,c4,c3,c4,ce,ad
748 7P DATA c0,cf,d3,e8,d6,c0,bf,d7,cd,cf
749 FO DATA d3,cc,c3,b7,d0,d7,c0,ba,b3,ce
750 1G DATA d8,d7,e0,db,c6,b8,bd,db,d4,c6
751 aB DATA b4,b8,b5,ce,d7,c3,b0,af,bb,e0
752 ML DATA d8,c6,c8,b3,b4,cf,d2,ca,c3,c5
753 Ji DATA c0,c5,d3,d4,d3,c2,bc,bf,e0,df
754 il DATA d0,c0,af,bf,d8,d1,c3,bf,b7,b0
755 B2 DATA c7,d7,d0,c3,b7,b8,cb,d0,de,d3
756 XK DATA b8,be,c5,d0,d3,bf,c0,bf,bf,c8
757 OI DATA cd,c3,cc,c1,cb,d3,cd,d0,cf,c9
758 7t DATA cc,bf,ba,cc,cf,c8,c7,bf,c4,d7
759 B9 DATA d0,d0,bd,b5,c6,cd,cc,cf,b9,b0
760 ru DATA cf,c9,c8,ca,c3,cc,c3,c2,ca,c5
761 I1 DATA c8,c7,c5,c4,cd,c7,c8,c2,cb,d6
762 nq DATA cd,ca,cb,cb,d6,d3,c4,ca,d3,cb
763 8F DATA ce,c9,ca,ca,bd,c8,d3,c5,c4,c5
764 HP DATA c0,c4,bf,c1,c4,bf,c0,c7,c6,c8
765 Ya DATA c3,c3,c8,c7,c5,c8,cf,d0,cf,c4
766 Pk DATA c0,d3,ca,c4,bd,cb,d0,bf,c2,cf
767 XO DATA c7,c4,c7,c1,c4,c9,c4,c8,c7,c2
768 vA DATA cf,cf,d0,cf,cb,c8,cb,c5,c4,bf
769 YI DATA c3,c8,c7,c4,c7,cb,d0,cf,d1,d0
770 Mc DATA cf,d5,d8,d2,cc,cd,c7,c8,cf,c7
771 gl DATA c8,c4,bb,c8,c9,c0,bc,b3,b0,b7
772 TW DATA bf,c8,bf,bd,c0,cb,d1,d8,ca,c0
773 HB DATA c7,cd,d0,cf,c5,c8,c7,c9,d4,d5
774 cE DATA d0,cf,c7,c8,cb,c7,ca,c9,bb,bc
775 ol DATA be,c6,cc,c9,c9,c3,c2,c4,c7,c2
776 5Z DATA c4,c3,ba,bf,c3,c9,cf,c7,c1,cb
777 ct DATA cf,d2,cf,c3,c6,c5,c9,cf,ce,cc
778 Zf DATA cb,c9,cc,cb,c7,cc,cb,c2,c6,c9
779 mr DATA c5,cb,c5,c4,c6,c5,cc,d3,cc,d2
780 Jb DATA cb,c6,cd,cb,c8,cd,c5,c4,c7,c9
781 w8 DATA d0,c7,c3,ca,c7,cc,d3,ce,cc,c6
782 S8 DATA c3,ca,c7,c8,ca,bf,c2,cd,cb,c8
783 5X DATA c5,c1,ca,cb,cc,d8,ca,c6,cf,cb
784 vO DATA c4,c7,c6,cc,ca,c6,ca,c5,c4,cb
785 z2 DATA c9,c4,c7,c3,c0,c3,c1,c4,bd,bb
786 tw DATA c6,c7,c4,c7,c9,c8,cb,c7,d0,cf
787 Ki DATA c9,d8,d3,d0,cf,cb,ca,cb,c9,d0
788 PZ DATA cd,c9,d0,ca,c0,c4,c2,c0,c3,bf
789 PN DATA c2,c7,c3,c8,c7,c3,cc,c7,c8,cd
790 kv DATA cb,d2,cd,cb,d0,cf,ce,d4,d5,d0
791 Fr DATA d5,cf,d0,d3,c5,c8,c7,c5,cc,c5
792 Qw DATA c0,c7,c2,c0,c3,bd,c0,bf,be,c2
793 Me DATA c3,c3,ca,c3,c8,cb,c5,ca,cb,c8
794 YA DATA cc,cb,c4,cc,cb,c8,cd,c5,c4,c9
795 yb DATA c7,d0,cf,cb,d0,ca,c4,cd,c7,c4
796 x9 DATA c7,c5,c8,cb,cb,d0,cd,c5,cc,cd
797 Nu DATA ca,d1,c9,cc,cd,cd,d0,cf,c9,ce
798 Wx DATA cb,c9,ce,c9,c8,cb,c7,cc,cd,cb
799 qZ DATA d0,cd,c8,cf,c7,c4,c7,c1,c2,c2
800 oI DATA bd,c2,bd,b9,c4,bf,c3,cb,c9,cc
801 EF DATA cd,c7,d0,ca,c8,cc,cb,cc,d3,d1
802 DW DATA d0,cf,cb,d0,d3,c9,ce,cb,c4,c7
803 S4 DATA c3,c4,c7,c1,c8,c9,c3,c8,c9,c5
804 Xp DATA cc,c3,c8,c7,bf,c4,c7,c6,ca,cb
805 2x DATA c9,ce,c9,c9,cf,ca,c8,cf,cb,cc
806 I8 DATA cf,cd,d0,cb,ca,d3,cf,ca,d5,d3
807 Sz DATA d0,ce,c7,cc,c7,c5,ca,bf,c0,bf
808 Mv DATA bd,c0,c3,c0,c8,c5,c0,c8,c3,c3
809 T3 DATA cb,c9,cc,cf,cb,d0,d2,cd,d4,ce
810 0g DATA cc,d5,cf,d2,d3,cf,d0,cf,c7,c8
811 2S DATA c6,c1,ca,c3,c4,c7,c1,c0,c5,c6
812 we DATA c8,c9,c1,c8,c3,c5,ce,c7,c8,cb
813 kC DATA c5,c8,c4,c4,c8,c5,c3,ce,cb,cc
814 9n DATA d3,d1,d2,d7,d4,dc,d9,d5,dc,d3
815 jI DATA d2,d7,d3,d8,d6,d2,d2,cf,cc,cc
816 kV DATA c6,c0,c7,bd,c0,bf,b7,c0,bb,b9
817 aj DATA c4,bf,c0,c6,be,c0,bf,be,c0,c3
818 6v DATA be,c8,c9,c8,ce,c7,ca,cf,c9,c9
819 jn DATA cd,c7,cc,cb,c8,d2,cf,d0,d7,d1
820 h4 DATA cc,cf,c9,d0,cd,c5,cc,c7,c8,cc
821 7A DATA c7,c8,cb,c7,c8,cb,c7,cc,cc,c9
822 jr DATA ce,c5,c1,c7,c3,37,00,00,00,00
823 HX DATA 00,00,00,00,00,00,00,00,00,00
824 IY DATA 00,00,00,00,00,00,00,00,00,00
825 JZ DATA 00,00,00,00,00,00,00,00,00,00
826 Ka DATA 00,00,00,00,00,00,00,00,00,00
827 Lb DATA 00,00,00,00,00,00,00,00,00,00
828 Mc DATA 00,00,00,00,00,00,00,00,00,00
829 Nd DATA 00,00,00,00,00,00,00,00,00,00
830 Oe DATA 00,00,00,00,00,00,00,00,00,00
831 Pf DATA 00,00,00,00,00,00,00,00,00,00
832 Qg DATA 00,00,00,00,00,00,00,00,00,00
833 wi DATA 00,00,03,ec,00,00,01,53,00,00
834 rR DATA 00,00,00,00,18,8e,00,00,18,80
835 Bp DATA 00,00,18,72,00,00,18,64,00,00
836 cx DATA 18,56,00,00,18,48,00,00,18,3a
837 Eh DATA 00,00,18,2c,00,00,18,1e,00,00
838 bC DATA 18,10,00,00,17,36,00,00,17,2c
839 SB DATA 00,00,17,24,00,00,17,1c,00,00
840 zQ DATA 17,12,00,00,17,08,00,00,16,f0
841 gH DATA 00,00,16,d0,00,00,16,c8,00,00
842 09 DATA 16,c2,00,00,16,b6,00,00,16,ae
843 IA DATA 00,00,16,a4,00,00,16,9c,00,00
844 SO DATA 16,94,00,00,16,8c,00,00,16,84
845 eC DATA 00,00,16,7c,00,00,16,74,00,00
846 b3 DATA 16,6a,00,00,16,62,00,00,16,5a
847 1v DATA 00,00,16,52,00,00,16,48,00,00
848 nN DATA 16,40,00,00,16,36,00,00,16,2c
849 SE DATA 00,00,16,24,00,00,16,1c,00,00
850 0h DATA 15,ea,00,00,15,e0,00,00,15,ce
851 G3 DATA 00,00,15,b0,00,00,15,7e,00,00
852 3r DATA 15,34,00,00,14,fc,00,00,14,e2
853 Er DATA 00,00,14,d2,00,00,14,c2,00,00
854 Gm DATA 14,b6,00,00,14,a6,00,00,14,a0
855 p7 DATA 00,00,14,9a,00,00,14,8e,00,00
856 89 DATA 14,86,00,00,14,7a,00,00,14,6e
857 2m DATA 00,00,14,66,00,00,14,5e,00,00
858 sp DATA 14,58,00,00,14,4a,00,00,14,34
859 1y DATA 00,00,14,2e,00,00,14,26,00,00
860 PY DATA 14,20,00,00,14,12,00,00,13,fa
861 kO DATA 00,00,13,f2,00,00,13,e6,00,00
862 Kv DATA 13,de,00,00,13,d8,00,00,13,d0
863 Jr DATA 00,00,13,c4,00,00,13,bc,00,00
864 j9 DATA 13,b4,00,00,13,ac,00,00,13,a6
865 1J DATA 00,00,13,9e,00,00,13,92,00,00
866 r0 DATA 13,8a,00,00,13,7e,00,00,13,68
867 cW DATA 00,00,13,62,00,00,13,52,00,00
868 eA DATA 13,4a,00,00,13,42,00,00,13,36
869 11 DATA 00,00,13,2e,00,00,13,26,00,00
870 0i DATA 13,04,00,00,12,fc,00,00,12,f4
871 6c DATA 00,00,12,ec,00,00,12,e4,00,00
872 Dv DATA 12,dc,00,00,12,a6,00,00,12,8c
873 8q DATA 00,00,12,86,00,00,12,7c,00,00
874 Gp DATA 12,70,00,00,12,66,00,00,12,5e
875 vq DATA 00,00,12,58,00,00,12,4e,00,00
876 pJ DATA 12,46,00,00,12,40,00,00,12,3a
877 HA DATA 00,00,12,34,00,00,12,2a,00,00
878 5G DATA 12,22,00,00,12,1c,00,00,12,16
879 7L DATA 00,00,12,10,00,00,12,06,00,00
880 pT DATA 11,fa,00,00,11,ee,00,00,11,e6
881 XI DATA 00,00,11,e0,00,00,11,d6,00,00
882 Me DATA 11,ce,00,00,11,c8,00,00,11,be
883 yb DATA 00,00,11,b2,00,00,11,aa,00,00
884 27 DATA 11,a4,00,00,11,9e,00,00,11,98
885 ZJ DATA 00,00,11,8e,00,00,11,82,00,00
886 GN DATA 11,76,00,00,11,6e,00,00,11,64
887 7x DATA 00,00,11,5a,00,00,11,44,00,00
888 x6 DATA 11,3e,00,00,11,34,00,00,11,26
889 UV DATA 00,00,11,1a,00,00,11,12,00,00
890 ht DATA 11,00,00,00,10,e8,00,00,10,d8
891 1m DATA 00,00,10,cc,00,00,10,b8,00,00
892 6D DATA 10,a2,00,00,10,96,00,00,10,8e
893 67 DATA 00,00,10,82,00,00,10,76,00,00
894 3G DATA 10,6a,00,00,10,5e,00,00,10,52
895 ik DATA 00,00,10,46,00,00,10,3e,00,00
896 nh DATA 10,2c,00,00,10,1a,00,00,10,08
897 mk DATA 00,00,0f,ec,00,00,0f,e4,00,00
898 fi DATA 0f,d2,00,00,0f,c0,00,00,0f,ae
899 hi DATA 00,00,0f,92,00,00,0f,8a,00,00
900 hV DATA 0f,78,00,00,0f,66,00,00,0f,54
901 wZ DATA 00,00,0f,38,00,00,0f,30,00,00
902 0X DATA 0f,1e,00,00,0f,0c,00,00,0e,f0
903 Pk DATA 00,00,0e,e8,00,00,0e,d6,00,00
904 1r DATA 0e,c4,00,00,0e,a8,00,00,0e,a0
905 ip DATA 00,00,0e,96,00,00,0e,7a,00,00
906 41 DATA 0e,64,00,00,0e,54,00,00,0e,36
907 Lr DATA 00,00,0e,18,00,00,0d,fc,00,00
908 JA DATA 0d,e0,00,00,0d,d8,00,00,0d,d0
909 t4 DATA 00,00,0d,c6,00,00,0d,bc,00,00
910 v4 DATA 0d,a6,00,00,0d,a0,00,00,0d,92
911 o6 DATA 00,00,0d,86,00,00,0d,7e,00,00
912 SJ DATA 0d,6c,00,00,0d,54,00,00,0d,44
913 in DATA 00,00,0d,38,00,00,0d,24,00,00
914 ln DATA 0d,0e,00,00,0d,02,00,00,0c,fa
915 Wd DATA 00,00,0c,ee,00,00,0c,e2,00,00
916 Ik DATA 0c,d6,00,00,0c,ca,00,00,0c,be
917 Sd DATA 00,00,0c,b2,00,00,0c,aa,00,00
918 rc DATA 0c,98,00,00,0c,86,00,00,0c,74
919 2i DATA 00,00,0c,58,00,00,0c,50,00,00
920 z0 DATA 0c,3e,00,00,0c,2c,00,00,0c,1a
921 r2 DATA 00,00,0b,fe,00,00,0b,f6,00,00
922 fd DATA 0b,e4,00,00,0b,d2,00,00,0b,c0
923 YG DATA 00,00,0b,a4,00,00,0b,9c,00,00
924 PC DATA 0b,8a,00,00,0b,78,00,00,0b,5c
925 vb DATA 00,00,0b,54,00,00,0b,42,00,00
926 aD DATA 0b,30,00,00,0b,14,00,00,0b,0c
927 wp DATA 00,00,0b,02,00,00,0a,e6,00,00
928 29 DATA 0a,d0,00,00,0a,c0,00,00,0a,a2
929 bG DATA 00,00,0a,84,00,00,0a,68,00,00
930 Jc DATA 0a,52,00,00,0a,4a,00,00,0a,3c
931 jL DATA 00,00,0a,34,00,00,0a,2c,00,00
932 wf DATA 0a,22,00,00,0a,18,00,00,0a,12
933 7L DATA 00,00,0a,04,00,00,09,fc,00,00
934 aq DATA 09,f2,00,00,09,e6,00,00,09,d2
935 ep DATA 00,00,09,ca,00,00,09,c2,00,00
936 7z DATA 09,ba,00,00,09,ae,00,00,09,a6
937 Zt DATA 00,00,09,8e,00,00,09,82,00,00
938 32 DATA 09,7a,00,00,09,66,00,00,09,36
939 y0 DATA 00,00,09,2e,00,00,09,1e,00,00
940 y2 DATA 09,16,00,00,09,06,00,00,08,fa
941 qF DATA 00,00,08,f2,00,00,08,e8,00,00
942 pN DATA 08,de,00,00,08,d2,00,00,08,ca
943 bw DATA 00,00,08,be,00,00,08,b2,00,00
944 yG DATA 08,a6,00,00,08,9e,00,00,08,96
945 Pi DATA 00,00,08,8a,00,00,08,82,00,00
946 Aq DATA 08,7a,00,00,08,6e,00,00,08,62
947 fC DATA 00,00,08,58,00,00,08,4a,00,00
948 XF DATA 08,40,00,00,08,36,00,00,08,2c
949 87 DATA 00,00,08,18,00,00,08,12,00,00
950 1c DATA 07,fa,00,00,07,f2,00,00,07,ea
951 hC DATA 00,00,07,e2,00,00,07,d8,00,00
952 K5 DATA 07,c2,00,00,07,ac,00,00,07,a4
953 4N DATA 00,00,07,92,00,00,07,7c,00,00
954 JM DATA 07,70,00,00,07,64,00,00,07,38
955 46 DATA 00,00,07,18,00,00,07,12,00,00
956 cF DATA 07,08,00,00,06,e4,00,00,06,be
957 SI DATA 00,00,06,ae,00,00,06,74,00,00
958 Uu DATA 06,54,00,00,06,4e,00,00,06,44
959 Js DATA 00,00,06,20,00,00,05,fa,00,00
960 em DATA 05,ea,00,00,05,44,00,00,05,1a
961 VY DATA 00,00,05,12,00,00,04,f8,00,00
962 2C DATA 04,f0,00,00,04,e8,00,00,04,e0
963 K3 DATA 00,00,04,d4,00,00,04,c8,00,00
964 A6 DATA 04,ae,00,00,04,94,00,00,04,7a
965 Q0 DATA 00,00,00,d2,00,00,00,cc,00,00
966 90 DATA 00,64,00,00,00,5e,00,00,00,58
967 dU DATA 00,00,00,52,00,00,00,4c,00,00
968 nd DATA 00,46,00,00,00,40,00,00,00,3a
969 GC DATA 00,00,00,34,00,00,00,1a,00,00
970 zw DATA 00,00,00,00,03,f2
```


Listing 1. (Schluß)

Krieg der Sterne

Es war einmal in ferner Zukunft... So wie der Film »Krieg der Sterne« könnte auch »Fantasy« beginnen; ein tolles Spiel, das in einer weit vor uns liegenden Zeit abläuft.

von Holger Hoffmann

Wir schreiben das Jahr 6533. In einer Welt aus 59 Planeten kämpfen einige Planeten um die Vorherrschaft. Während auf den Heimatplaneten die Produktion auf Hochtouren läuft, beginnen die Herrscher andere Planeten zu erobern. Lassen Sie sich von »Fantasy« in diese Welt entführen. Aber nur, wenn Sie genügend Zeit haben, denn Fantasy wird Sie lange fesseln.

Nach dem Abtippen (siehe Kasten mit Hinweisen), kann das Abenteuer für ein bis vier Spieler beginnen. Fantasy wird, bis auf die Eingabe von Zahlen und Namen, komplett mit der Maus gesteuert. Am rechten Bildschirm finden Sie jeweils die Menüpunkte in Form von Schaltern.

■ Spiel laden: Hiermit laden Sie ein vorher gespeichertes Spiel von Diskette. Der Dateiname wird mit Pfad (z.B. DF1:Welten/Welt_2) eingegeben. Anschließend beginnt das Spiel.

■ Neues starten: Fantasy kann mehrere Welten verwalten. Für jede Welt besteht die Möglichkeit, mehrere Spiele zu speichern. Mit diesem Menüpunkt starten Sie ein neues Spiel auf einer schon generierten Welt (siehe auch »Welt kreieren«). Nach dem Anwählen erscheint ein Fenster, in dem Sie nun den Namen der gewünschten Welt eingeben. Danach müssen Sie noch die Fragen nach der Zahl und den Namen der Spieler beantworten.

■ Welt kreieren: Sie können mehrere verschiedene Welten anlegen. Nach Anwahl dieses Schalters zeichnet Fantasy dreimal ein Dreieck in die linke obere Ecke des Bildschirms. Nach diesem Vorgang geben Sie den Namen zur Speicherung auf Diskette ein.

■ PRG - beenden: Dient zum Verlassen des Spiels und des Basic-Interpreters. Wenn ein Spiel geladen oder ein neues gestartet wurde, erscheint das eigentliche Spielfeld.

■ Bauen: Es erscheint ein Untermenü mit neuen Schaltern:

- Kriegsschiffe: Pro Schiff müssen auf dem Heimatplaneten (in der Grafik von einem Kreis umgeben) drei Geldeinheiten sowie vier Rohstoffeinheiten vorhanden sein. Sie erhalten die neuen Schiffe in der nächsten Runde.
- Transportschiffe: Zum Bau benötigt man pro Schiff eine Geld- und drei Rohstoffeinheiten.
- Gewerbezentrum: Um von einem Planeten Rohstoffe abtransportieren zu können, muß man bestimmte Lizenzen zahlen und Verladeanlagen einrichten. Das kostet 50 Geldeinheiten.
- Industriezentrum: Für die »lächerliche« Summe von 500 Geldeinheiten können Sie Ihren Heimatplaneten (Produktion) verlegen.
- Weiter
  Dient zur Rückkehr ins Hauptmenü.

■ Angriff: Hier zeigt sich die komfortable Seite von Fantasy. Nach dem Anklicken des Schalters wählen Sie den Planeten, von dem Sie Schiffe losschicken wollen, per Mausklick an. Danach bestimmen Sie den Zielplaneten ebenfalls durch Anklicken. Nun verlangt Fantasy die Eingabe, wie viele Schiffe starten sollen. Bevor Sie versuchen einen Planeten zu erobern, sollten Sie ein einzelnes Schiff losschicken. Ist keine Bevölkerung vorhanden, besetzen Sie den Planeten sofort. Andernfalls baut die Besatzung einen Spionagering auf. Ob der Angriff erfolgreich war, zeigt sich in der nächsten Spielrunde.

■ Transport: Genau wie bei den Angriffen, legen Sie den Start- und Zielplaneten durch Mausklick fest. Zusätzlich zur Anzahl der zu verwendenden Schiffe, geben Sie noch die Zahl der zu transportierenden Rohstoffe ein. Pro Schiff ist eine Einheit möglich. Wichtig ist, daß beide Planeten Ihnen gehören müssen.

■ Staats Info: Das Programm zeigt wichtige Daten über Ihr Imperium an. Bei der Zahl der Schiffe ist zu beachten, daß Fantasy die in diesem Spielzug schon bewegten Schiffe nicht berücksichtigt.

■ Planeten Info: Nach Anklicken eines Planeten erhalten Sie Informationen.

■ Grafik: Natürlich kann sich niemand merken, wieviel Schiffe und Rohstoffe wo zu finden sind. Darum zeigt Ihnen Fantasy solche Informationen grafisch an. Im Untermenü gibt es folgende Punkte:

- Entfernungen: Durch mehrmaliges Anklicken des Schalters wählen Sie die drei Einstellungen.
- Lager: Alle Planeten mit über 30 Einheiten an Rohstoff werden blau eingefärbt. Bei mehr als 150 Einheiten wird er grau dargestellt.
- Transportsch.: Blau bedeutet mehr als 10, grau mehr als 50 Transportschiffe auf dem Planeten. Ansonsten ist der Planet braun.
- Kriegsschiffe: Hier gelten dieselben Angaben wie bei den Transportschiffen.
- Planet: Bei hohem Steuereinkommen ist der Planet blau, bei hohem Rohstoffvorkommen grau.
- Weiter: Rückkehr ins Hauptmenü.

■ Revolution: Hin und wieder finden auf den Planeten Revolutionen statt. Um auf einem fremden Planeten eine Revolution zu begünstigen, können Sie Geld investieren. Klicken Sie dazu einfach auf den entsprechenden Planeten. Gelingt die Revolution, gehört der Planet anschließend Ihnen. Damit ein Aufstand überhaupt möglich ist, muß sich ein Spionagering auf dem Planeten aufhalten (siehe auch »Angriff«). Pro Runde findet nur ein Aufstand statt. Wählen mehrere Spieler diesen Punkt an, so findet der zweite eine Runde später statt. Noch eine schlechte Nachricht: Auch auf den

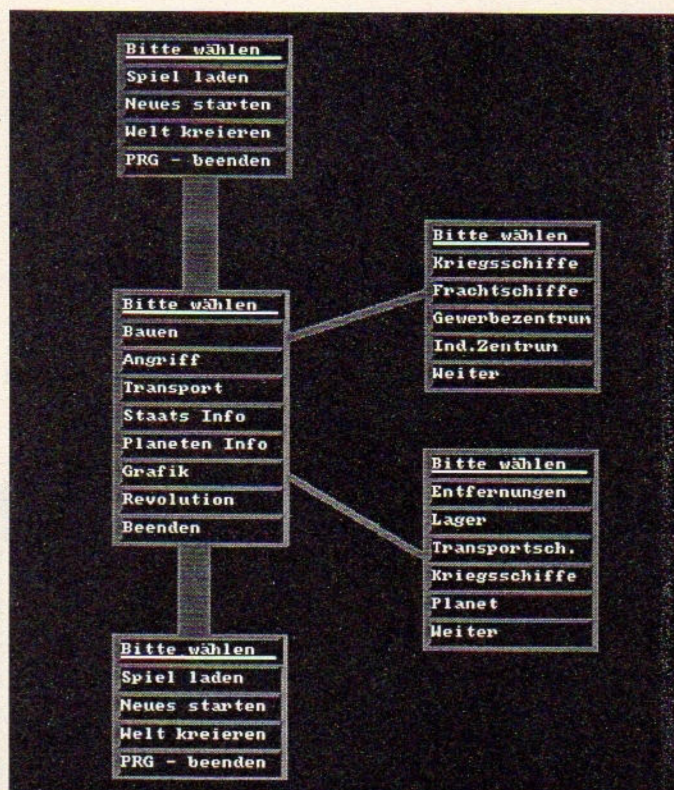


Bild 1. Die Menüstruktur von »Fantasy«. Drei Untermenüs gibt es. Das unten stehende Menü erreichen Sie über »Beenden«.

Holger Hoffmann



Anfang 1984 stieg Holger Hoffmann in den C64-Boom mit ein. Nach wenigen Wochen erfolgte der Umstieg von Basic auf Assembler. Nach dem Preissturz des Amiga legte sich der Autor einen Amiga 500 zu. Auch hier erfolgte der Start in Basic. Neben Assembler arbeitet Holger Hoffmann noch mit C und Modula-2. Sein Studienwunsch ist Mathematik-Informatik, sein nächster Hardware-Wunsch eine Festplatte.

von Ihnen besetzten Planeten kann ein Aufstand stattfinden. Ist er erfolgreich, verlieren Sie den Planeten.

■ Beenden: Im Mehrspielermodus übergeben Sie hiermit die Kontrolle an den nächsten Herrscher. Damit alle anderen Mitspieler wegsehen können, erscheint anschließend ein Fenster, in dem der nächste Herrscher genannt wird. Nach dem letzten Spieler erscheint ein Menü, in dem man laden und speichern kann. Bei nur einem Spieler erscheint dieses Menü sofort.

■ Save Game: Speichert den aktuellen Spielstand.

■ Load Game: Lädt ein vorher gespeichertes Spiel.

■ Ende: Führt zu dem am Anfang erklärten Menü zurück.

■ Weiter: Die nächste Spielrunde wird gestartet.

Das Spiel endet, wenn nur noch ein Herrscher seinen Heimatplaneten besitzt. Der Einspielermodus dient nur zu Trainingszwecken und endet nicht. Aber auch alleine macht Fantasy viel Spaß und außerdem lernt man so Tricks und Kniffe, damit man beim nächsten Turnier der Stärkere ist. *kn*

Hinweise zum Abtippen

Damit Sie Fantasy spielen können, müssen einige Vorbedingungen erfüllt sein:

- Folgende Zeichensätze müssen vorhanden sein:
  `diamond 20` `opal 9` `sapphire 14`
  Dies ist auf alle Fälle gewährleistet, wenn Sie von Ihrer Original-Workbench-Diskette starten.
- Die folgenden Dateien müssen im selben Verzeichnis stehen wie Amiga-Basic und das Programm »Fantasy«:
  `intuition.bmap` `graphics.bmap` `diskfont.bmap`

Fantasy benötigt viel Speicher. Bei Amigas mit 512 KByte RAM kann das zu Problemen führen. Diese lassen sich jedoch umgehen. Schalten Sie den Computer aus, damit erreicht man, daß nach dem Einschalten der gesamte Speicher zur Verfügung steht. Schließen Sie alle CLI-Fenster. Starten Sie das Programm von der Workbench aus durch Doppelklick auf das Programmsymbol. Starten Sie kein anderes Programm. Auf keinen Fall dürfen Sie das CLI-Kommando ADDBUFFERS einsetzen.

Programmname:	Fantasy
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Basic
Bemerkung:	siehe Kasten

```
Programmautor: Holger Hoffmann
------------------------------
NxO CLEAR,52000&
8I  DECLARE FUNCTION opendiskfont& LIBRARY
mA  DECLARE FUNCTION openfont& LIBRARY
qI  DECLARE FUNCTION AskSoftStyle& LIBRARY
1h  DECLARE FUNCTION SetSoftStyle& LIBRARY
HU  LIBRARY "intuition.library"
17  LIBRARY "graphics.library"
px  LIBRARY "diskfont.library"
vF  GOSUB font:GOSUB titel:RANDOMIZE TIMER:GOSUB ssr
AI  FOR x= 0 TO 7:PALETTE x,0,0,0:NEXT
mo  DIM a%(828),g1(4),g2(4),s1(4),s2(4)
oW  DIM kx(59),ky(59),k1(4,50),k2(4,50),k3(4,50),de1(4,50),he1(
    4,50),de2(4,50)
Pt  DIM ro(59),mo(59),la(59),sh(59),tr(59)
tQ  DIM na(59),st(4,59),na$(4),f2(4),f3(4),f4(4),f5(4),v(4)
KL  DIM a1(20),a2(20),a3(20),a4(20),re(59)
rU  DIM st$(5),ve(600),ha(3),heim(4),f1(4),gf&(3200)
1H  st$(0)="Existenz bekannt":st$(1)="bekannt"
o0  st$(2)="besitzend":st$(3)="gewerblich nutzend"
AX  st$(4)="industriell nutzend"
B7  sr(1)=31:sr(2)=35:sr(3)=38
Gs  f2(0)=2:f3(0)=1:GOSUB planeten:CLS
JZ  na$(0)="Ureinwohner"
Uw  neus:PALETTE 0,0,0,0
qd  PALETTE 1,1,1,1:PALETTE 2,0,0,.6:PALETTE 3,.5,.5,1
WN  PALETTE 4,0,0,1:PALETTE 5,.7,.3,0
iz  PALETTE 6,.5,.5,.5:PALETTE 7,1,0,0
qj  ON ERROR GOTO fehler
y2  start:
yl  CLS:GOSUB menue1
xW  ww5:
qu  GOSUB eingabe
M3  IF wa=1 THEN GOSUB lod :IF s$<>"" THEN anfang:ELSE GOTO w
    w5
er  IF wa=2 THEN GOTO neu
bx  IF wa=3 THEN GOSUB grafik:GOTO start
yD  IF wa=4 THEN MENU RESET:GOSUB cfont:WINDOW CLOSE 1:SCREEN C
    LOSE 1:END
dl  GOTO ww5
2C  neu:
Kx  GOSUB wlod:IF f$="" THEN start
1A  GOSUB zufall:GOSUB wieviele
nv  jahr=1
OP  anfang:
BI  FOR l= 1 TO spieler
6I2   IF na(heim(l))<>l THEN ww3
Kx    g1(l)=0:g2(l)=0:s1(l)=0:s2(l)=0
4W    na$="Runde :"+STR$(jahr)
9h0 ww4:
Ca2   IF LEN(na$)<12 THEN na$=na$+" ":GOTO ww4
mR    na$=na$+"                 "+na$(l)+CHR$(0)
OY    CALL SetWindowTitles(WINDOW(7),SADD(na$),0)
Tu    GOSUB erf
MA    CLS:GOSUB menue2
WP    IF spieler<>1 THEN GOSUB nex
M8    di=f1(l):GOSUB dis
9f0 ww2:
EI2   GOSUB eingabe
pl    IF m = 8 THEN GOTO ww3
Wc    IF m=4 THEN GOSUB einkommen
dV    IF m=1 THEN GOSUB bauen
Eb    IF m=6 THEN GOSUB graf
My    IF (z<>0) AND (wa<>0) THEN ON wa GOSUB bauen,senden,t
      ransi,info,info,info,revo,info,info
qv    GOTO ww2
Ls0 ww3:
wg2   p=heim(l):sh(p)=sh(p)+g1(l):tr(p)=tr(p)+g2(l)
Zs0 NEXT l
3C  jahr=jahr+1
HO  COLOR 1,0:CLS:FOR p = 1 TO 59:la=na(p)
u22   IF la<>0 AND st(la,p)>1 THEN v(la)=v(la)+mo(p):la(p)=l
      a(p)+ro(p)
BG0 NEXT
bR  na$="Fantasy  Written in 1988 by Holger Hoffmann"+CHR$(0)
Jt  CALL SetWindowTitles(WINDOW(7),SADD(na$),0)
Uc  FOR l = 1 TO spieler
H12   IF s1(l)=0 THEN ww7
On    FOR w=0 TO s1(l)-1:GOSUB war:NEXT w
v2    s1(l)=0
oP0 ww7:
WD2   IF s2(l)=0 THEN ww8
2c    FOR w=0 TO s2(l)-1
s34     p= de2(l,w):la(p)=la(p)+k3(l,w)
jd      tr(p)=tr(p)+k2(l,w)
Bf2   NEXT w
5D0 s2(l)=0
zb  ww8:
sB  NEXT l
MZ2   pl=INT(RND(1)*58+1)
AZ    FOR x= 1 TO 59:IF na(x)=0 THEN sh(x)=sh(x)+INT((mo(x)+ro(
      x))/10)
Er4     IF re(x)>re(pl) THEN pl=x
Kp2   NEXT x
t9    IF re(pl)=0 THEN zu=RND(1) ELSE zu=1
4p    h=0:FOR x=1 TO spieler:IF pl=heim(x) THEN h=1
Ns    NEXT x
tN    GOSUB menue7
YJ    IF (zu>.4) THEN GOSUB rev
2c0 ww6:
rv2   GOSUB eingabe
dM    IF wa=1 THEN GOSUB sav
bJ    IF wa=2 THEN GOSUB lod
st    IF wa=3 THEN beenden
4a    IF wa<>4 THEN ww6
lr    en=0
T4    FOR x = 1 TO spieler:IF na(heim(x))=x THEN en=en+1
Y3    NEXT x
d3    IF (spieler=1) OR (en>1) THEN GOTO anfang
48    SCREEN CLOSE 1:WINDOW CLOSE 1
kV    GOTO sieg
Xf0 dars:
```

Bitte mit dem Checksummer (Seite 143) eingeben. Der Checksummer und dieses Listing befinden sich auf der Programmservice-Diskette dieser Ausgabe (siehe auch Seite 34)


```
di=di+1:IF di=4 THEN di=1
GOSUB dis:RETURN
planeten:
DATA 0,0,0,0,0,0,5,5,5,5,0,0,0,0,0,0
DATA 0,0,0,5,5,7,5,5,5,5,0,5,5,0,0,0
DATA 0,5,5,7,5,5,7,5,5,5,0,5,5,0,5,0
DATA 5,5,7,5,5,7,5,5,0,5,5,0,5,5,5,0
DATA 5,7,5,5,7,5,5,7,5,5,0,5,0,5,0,5
DATA 0,5,7,5,5,5,7,5,5,5,5,5,5,0,5,0
DATA 0,0,0,5,5,7,5,7,5,5,5,0,5,0,0,0
DATA 0,0,0,0,0,0,7,5,5,5,0,0,0,0,0,0
DATA 0,0,0,0,0,0,4,4,4,4,0,0,0,0,0,0
DATA 0,0,0,4,4,4,3,4,4,4,2,4,4,0,0,0
DATA 0,3,4,3,4,4,4,4,4,4,4,4,4,2,4,0
DATA 3,4,3,4,3,4,4,4,4,2,4,2,2,4,2,4
DATA 3,3,4,3,4,4,4,4,4,4,4,2,4,2,4,2
DATA 0,3,4,4,4,3,4,4,4,4,2,4,2,4,4,0
DATA 0,0,0,4,3,4,4,4,4,4,4,2,4,0,0,0
DATA 0,0,0,0,0,0,4,4,4,4,0,0,0,0,0,0
DATA 0,0,0,0,0,0,6,6,6,6,0,0,0,0,0,0
DATA 0,0,0,6,6,6,1,6,6,6,0,6,6,0,0,0
DATA 0,1,6,1,6,6,6,6,6,6,6,6,6,0,6,0
DATA 1,6,1,6,1,6,1,6,6,0,6,0,6,6,0,6
DATA 1,1,6,1,6,6,6,6,6,6,6,6,6,6,6,6
DATA 0,1,6,6,6,1,6,6,6,6,0,6,0,6,6,0
DATA 0,0,0,6,1,6,6,6,6,6,6,6,6,0,0,0
DATA 0,0,0,0,0,0,6,6,6,6,0,0,0,0,0,0
DATA 0,0,0,0,0,0,7,7,7,7,0,0,0,0,0,0
DATA 0,0,0,7,7,7,0,0,0,0,7,7,7,0,0,0
DATA 0,7,7,7,0,0,0,0,0,0,0,0,7,7,7,0
DATA 7,7,0,0,0,0,0,0,0,0,0,0,0,0,7,7
DATA 7,7,0,0,0,0,0,0,0,0,0,0,0,0,7,7
DATA 0,7,7,7,0,0,0,0,0,0,0,0,7,7,7,0
DATA 0,0,0,7,7,7,0,0,0,0,7,7,7,0,0,0
DATA 0,0,0,0,0,0,7,7,7,7,0,0,0,0,0,0
FOR s = 0 TO 3
  FOR y= 1 TO 8
    FOR x= 1 TO 16
      READ a:PSET(16*s+x,y),a
NEXT x,y,s
GET (1,1)-(16,8),a1:GET (17,1)-(32,8),a2
GET (33,1)-(48,8),a3:GET (49,1)-(64,8),a4
RETURN
grafik:
COLOR 1,0:CLS:z=1
FOR x= 30 TO 450 STEP 70
  FOR y = 20 TO 220 STEP 50
    kx(z)=x+(RND(1)*18)-9:ky(z)=y+(RND(1)*18)-9
    st(1,z)=0:z=z+1
NEXT y,x
FOR x= 65 TO 415 STEP 70
  FOR y = 45 TO 195 STEP 50
    kx(z)=x+(RND(1)*26)-13:ky(z)=y+(RND(1)*26)-13
    st(1,z)=0:z=z+1
NEXT y,x
hh=1:l=1:sp2=0:sp1=26
FOR en = 1 TO 3:sp1=sp1+(6-en)
  FOR x= 1 TO 59: FOR y= x TO 59
    PSET(x,y),en
    ss=SQR(((kx(x)-kx(y))/2)^2+(ky(x)-ky(y))^2)
    IF (ss<sp1) AND (ss>sp2) THEN ve(hh)=x:ve(hh+1)=y:hh=
    hh+2:
  NEXT y,x
  ha(en)=hh-2
NEXT en
GOSUB wsav:RETURN
puty:
FOR h= 1 TO 59
  IF POINT(kx(h),ky(h))=0 THEN put1
  IF st(1,h)<2 THEN PUT (kx(h)-8,ky(h)-4),a4,PSET:GOTO put
  1
  IF ro(h)>15 THEN PUT (kx(h)-8,ky(h)-4),a3,PSET:GOTO put1
  IF mo(h)>8 THEN PUT (kx(h)-8,ky(h)-4),a2,PSET:GOTO put1
  PUT (kx(h)-8,ky(h)-4),a1,PSET
put1:
NEXT h
RETURN
menue1:
mm$(1)="Spiel laden   ":mm$(2)="Neues starten "
mm$(3)="Welt kreieren ":mm$(4)="PRG - beenden "
me = 4:GOSUB menue:RETURN
menue2:
mm$(1)="Bauen         ":mm$(2)="Angriff       "
mm$(3)="Transport     ":mm$(4)="Staats Info   "
mm$(5)="Planeten Info ":mm$(6)="Grafik        "
mm$(7)="Revolution    ":mm$(8)="Beenden       "
me = 8:GOSUB menue:RETURN
menue3:
mm$(1)="Spiel laden   ":mm$(2)="Spiel sichern "
me = 2:GOSUB menue:RETURN
menue4:
mm$(1)="Neuanfang     ":mm$(2)="In Ordnung    "
mm$(3)="Beenden       "
me=3:GOSUB menue:RETURN
menue5:
mm$(1)="Kriegsschiffe ":mm$(2)="Frachtschiffe "
mm$(3)="Gewerbezentrum":mm$(4)="Ind.Zentrum   "
mm$(5)="Weiter        "
me=5:GOSUB menue:RETURN
menue7:
mm$(1)="Save Game     ":mm$(2)="Load Game     "
mm$(3)="Ende          ":mm$(4)="Weiter        "
me=4: GOSUB menue:RETURN
menue8:
mm$(1)="Entfernungen  ":mm$(2)="Lager         "
mm$(3)="Transportsch. ":mm$(4)="Kriegsschiffe "
mm$(5)="Planet        ":mm$(6)="Weiter        "
me=6: GOSUB menue:RETURN
menue:
COLOR 1,4
LINE (506,2)-(630,240),7,bf:LINE (509,4)-(627,20),3,bf
LINE (512,6)-(624,18),4,bf:LINE (510,20)-(626,20),2
LINE (511,19)-(626,19),2:LINE (625,20)-(625,5),2
LINE (626,20)-(626,4),2:LINE (627,20)-(627,4),2
FOR t= 1 TO me
  LOCATE 2,65: PRINT mm$(t)
  GET (510,4)-(626,20),a%
  LINE (510,4+t*18)-(626,20+t*18),0,bf
  PUT (510,4+t*18),a%
NEXT t
LOCATE 2,65: PRINT "Bitte wählen  "
wa=0:m=0:COLOR 1,0
LINE (512,16+wa*18)-(622,17+wa*18),1,bf
RETURN
zufall:
FOR z = 1 TO 59:zu=RND(1)
  IF zu >.75 THEN ro(z)=INT(RND(1)*10)+15 ELSE ro(z)=1+INT
  (RND(1)*10)
  IF zu <.15 THEN mo(z)=INT(RND(1)*8)+8 ELSE mo(z)=1+INT(R
  ND(1)*6)+2
  sh(z)=INT((mo(z)+ro(z)*2)*(RND(1)/5+.9))
  tr(z)=0:na(z)=0:re(pl)=0:la(pl)=0
  st(1,z)=0:st(2,z)=0:st(3,z)=0:st(4,z)=0
NEXT z
RETURN
eingabe:
n=MOUSE(0)
IF n<=0 THEN eingabe
y=MOUSE(2):x=MOUSE(1)
IF x>507 THEN ein2
z=0
FOR h= 1 TO 59
  IF x<kx(h)+10 AND x>kx(h)-10 AND y<ky(h)+5 AND y>ky(h
  )-5 THEN m=0:z=h:h=86
NEXT h
IF z= 0 THEN eingabe ELSE GOTO ein3:
ein2:
z=0:m=INT((y-4)/18):IF (m>me) OR (m<0) THEN GOTO eingabe
LINE (512,16+wa*18)-(622,17+wa*18),4,bf: LINE (512,16+m*18)
-(622,17+m*18),1,bf:wa=m
ein3:
RETURN
wieviele:
COLOR 1,0: WINDOW 2,"",(200,100)-(440,150),2,1
PRINT:PRINT " - Wieviele Mitspieler -"
PRINT:PRINT"          (1-4) ";:COLOR 7,0:INPUT spieler
IF spieler <1 THEN spieler = 1
IF spieler >4 THEN spieler = 4
spieler = INT(spieler)
FOR h = 1 TO spieler
  COLOR 1,0: CLS:PRINT
  PRINT "    Name des ";h;". Spielers"
  PRINT
```

Listing. »Fantasy« versetzt Sie in das Jahr 6533. Erobern Sie neue Welten.

```
263 OW    PRINT "     ";:COLOR 7,0:INPUT na$(h)
264 wE0 wie1:
265 eq2   zu=1+INT(RND(1)*59)
266 vy    IF na(zu)<>0 THEN wie1
267 Wk    za=0
268 mp    FOR r= 1 TO ha(1):IF ve(r)=zu THEN za=za+1
269 4T    NEXT r
270 VT    IF za<3 THEN wie1
271 dy    na(zu)=h:st(h,zu)=4:heim(h)=zu:sh(zu)=40
272 tV    IF ro(zu)<15 THEN la(zu)=60:ELSE la(zu)=ro(zu)*2:sh(zu)=
          10
273 F0    IF mo(zu)<10 THEN v(h)=80:ELSE v(h)=30:sh(zu)=30
274 Ag    f1(h)=1:f2(h)=2:f3(h)=1:f4(h)=10+INT(RND(1)*5):f5(h)=1
275 VU    IF ro(zu)<8 THEN ro(zu)=8
276 XS    IF mo(zu)<6 THEN mo(zu)=6
277 s70 NEXT h
278 AX  WINDOW CLOSE 2: RETURN
279 UW  senden:
280 yC  IF st(1,z)<2 OR s1(1)>50 THEN BEEP:RETURN
281 s1  COLOR 1,7:LINE (64*8,24*8)-(630,28*8),7,bf
282 Oi  LOCATE 25,65:PRINT "Schiffe senden";
283 6C  LOCATE 26,65:PRINT "von NR.";z
284 fA  LOCATE 27,65:PRINT "zu NR.";
285 S6  z1=z
286 x1  GOSUB eingabe
287 xa  IF (z=0) OR (z=z1)THEN LINE (64*8,24*8)-(630,28*8),7,bf:RET
        URN
288 8n  PRINT z:COLOR 1,0
289 re  er=SQR(((kx(z)-kx(z1))/2)^2+(ky(z)-ky(z1))^2)
290 gZ  IF er > sr(f1(1)) THEN BEEP:RETURN
291 M2  GET (146,89)-(454,182),gf&
292 rY  WINDOW 2,"Schiffe senden :",(150,100)-(450,180),0,1
293 NZ  n$=" Von Planet NR.":GOSUB text:PRINT z1
294 12  PRINT "  zu Planet NR.";z:PRINT
295 wH  PRINT "  Stationierte Kriegsschiffe :";sh(z1):PRINT
296 10  PRINT "  Wieviele ":PRINT "  Kriegsschiffe  ";
297 I7  GOSUB of:INPUT a1:IF a1<0 THEN a1=0
298 8L  IF a1>sh(z1) THEN a1=sh(z1)
299 dh  de1(1,s1(1))=z:he1(1,s1(1))=z1:k1(1,s1(1))=a1
300 94  sh(z1)=sh(z1)-a1:s1(1)=s1(1)+1
301 k1  WINDOW CLOSE 2:PUT (146,89),gf&
302 Eq  RETURN
303 ba  bauen:
304 EY  p=heim(1):IF na(p)<>1 THEN RETURN
305 Fh  GOSUB menue5
306 af  bau1:
307 IM  GOSUB eingabe
308 hn  bau2:
309 2G  IF wa=5 THEN GOSUB menue2:RETURN
310 71  ON wa GOTO krieg,trans,hand,indu
311 pZ  GOTO bau1
312 Xs  krieg:GET (146,89)-(404,192),gf&
313 F3  WINDOW 2,"Kriegsschiffe bauen :",(150,100)-(400,190),0,1
314 Ph  f1=INT(la(p)/4):f2=INT(v(1)/3)
315 Yg  GOSUB bau3
316 EU  la(p)=la(p)-za*4:v(1)=v(1)-za*3:g1(1)=g1(1)+za
317 OW  WINDOW CLOSE 2:PUT (146,89),gf&:GOTO bau1
318 AD  trans:
319 kH  GET (146,89)-(404,192),gf&
320 r2  WINDOW 2,"Transportschiffe bauen :",(150,100)-(400,190),0,1
321 e6  f1=INT(la(p)/3):f2=INT(v(1))
322 fn  GOSUB bau3
323 Ow  la(p)=la(p)-za*3:v(1)=v(1)-za:g2(1)=g2(1)+za
324 Vd  WINDOW CLOSE 2:PUT (146,89),gf&:GOTO bau1
325 Hs  bau3:n$=" Sie haben":GOSUB text:PRINT
326 vR  PRINT"  Geld für ";f2;" Schiffe und "
327 nJ  PRINT "  Rohstoffe für ";f1;" Schiffe."
328 ZK  PRINT:PRINT "  Wieviele Schiffe wollen Sie"
329 T8  PRINT "  bauen ";
330 Zv  GOSUB of:INPUT za:IF za<0 THEN za=0
331 YS  IF (za=<f1) AND (za=<f2) THEN RETURN
332 fn  GOSUB sf:PRINT "  Nicht genug Mittel !":PRINT:BEEP
333 rI  x=70:y=70:GOSUB ok:za=0
334 kM  RETURN
335 MC  einkommen:
336 h6  GET (196,69)-(408,216),gf&
337 XO  WINDOW 2,"Staats Info",(200,80)-(404,214),0,1
338 wG  C=0:t=0:g=0:a=0:q=0
339 I7  FOR h=1 TO 59
340 BT2   IF na(h)=1 THEN g=g+mo(h):q=q+ro(h):a=a+la(h):C=C+sh(h):t
          =t+tr(h):
341 u90 NEXT h
342 Sb  n$=" Besitz :":GOSUB text
343 xT  PRINT:PRINT:PRINT " Steuereinkommen :";g
344 KI  PRINT " Vermoegen        :";v(1):PRINT " Rohstoffabbau   :";
        q
345 RB  PRINT " Rohstoffe         :";a:PRINT " Kriegsschiffe     :";
        C
346 4J  PRINT " Transportschiffe :";t:PRINT
347 nh  PRINT " Reichweite     :";sr(f1(1)):PRINT " Bewaffnung   :";
        f2(1)
348 Jn  PRINT " Panzerung     :";f3(1)
349 EP  x=43:y=115:GOSUB ok
350 ao  WINDOW CLOSE 2:PUT (196,69),gf&
351 1d  RETURN
352 DN  info:
353 ed  GET (176,89)-(420,220),gf&
354 S7  WINDOW 2,"Planeten Info",(180,100)-(416,218),0,1
355 ln  PRINT:n$= " Planet NR.":GOSUB text:PRINT z
356 79  PRINT:PRINT "  Besitzer: ";na$(na(z)):PRINT
357 Xq  s=st(1,z):PRINT "  Status: ";st$(s)
358 mH  IF s=0 THEN infoend
359 Us  PRINT "  Kriegsschiffe :";sh(z):PRINT "  Einkommen :";mo(z)
360 fy  PRINT "  Rohstoffe :";ro(z)
361 6t  IF  s=1 THEN infoend
362 5k  PRINT "  Lager     :";la(z):PRINT "  Transportschiffe :";tr
        (z)
363 2C  infoend:
364 Ks  x=61:y=99
365 wI  GOSUB ok
366 w0  WINDOW CLOSE 2:PUT (176,89),gf&
367 Ht  RETURN
368 Pu  wsav:
369 9H  WINDOW 2,"Welt abspeichern",(200,100)-(440,150),2,1
370 AZ  PRINT:PRINT "  Bitte geben sie"
371 it  PRINT "  einen Namen ein":INPUT "  ";a$
372 fm  OPEN a$ FOR OUTPUT AS #1
373 gn  PRINT#1,ha(1),ha(2),ha(3)
374 Cn  FOR x = 1 TO ha(3)+1:PRINT#1,ve(x):NEXT x
375 nB  FOR x= 1 TO 59:PRINT#1,kx(x),ky(x):NEXT x
376 Vd  CLOSE 1:WINDOW CLOSE 2
377 R3  RETURN
378 g0  wlod:
379 oC  WINDOW 2,"Welt aussuchen",(200,100)-(440,150),2,1
380 re  CLS:PRINT:PRINT "  Bitte geben sie den"
381 4k  PRINT "  Namen der Welt ein,":PRINT "  die sie benutzen wol
        len"
382 v8  INPUT "  ";f$
383 um  IF f$<>"" THEN GOSUB wwlod
384 Gs  WINDOW CLOSE 2:RETURN
385 vG  wwlod:
386 uv  OPEN f$ FOR INPUT AS #1
387 Vf  INPUT#1,ha(1),ha(2),ha(3)
388 49  FOR x = 1 TO ha(3)+1:INPUT#1,ve(x):NEXT x
389 YU  FOR x = 1 TO 59:INPUT#1,kx(x),ky(x):NEXT x
390 6q  CLOSE 1
391 fH  RETURN
392 fr  war:
393 W1  p=de1(1,w)
394 X4  IF na(p)=1 THEN sh(p)=sh(p)+k1(1,w):RETURN
395 xB  kr1:
396 00  t1=k1(1,w):IF t1=0 THEN RETURN
397 Eb  t2=sh(p)
398 4J  kr2:
399 Lc  ff1=t1*((15-RND(1)*10)*(f2(1)-f3(1)))/100 :IF ff1<0 THEN f
        f1=0
400 RD  ff2=t2*((15-RND(1)*10)*(f2(na(p))-f3(na(p))))/100 :IF ff2<
        0 THEN ff2=0
401 f3  t1=INT(t1-ff2):IF t1<0 THEN t1=0
402 wD  t2=INT(t2-ff1):IF t2<0 THEN t2=0
403 uK  IF t1<t2 THEN ver1
404 RV  IF t2<>0 THEN kr2
405 Ru  sh(p)=t1:st(na(p),p)=1:na(p)=1
406 rS  st(1,p)=2:IF p=heim(1) THEN st(1,p)=4
407 vX  RETURN
408 zy  ver1:IF na(he1(1,w))=1 THEN sh(he1(1,w))=sh(he1(1,w))+t1
409 fI  sh(p)=t2:st (1,p)=1
410 ya  RETURN
411 Rf  transi:
412 zs  COLOR 1,7:LINE (64*8,24*8)-(630,28*8),7,bf
413 jz  LOCATE 25,65:PRINT "Transport ";
414 DJ  LOCATE 26,65:PRINT "von NR.";z
415 mH  LOCATE 27,65:PRINT "zu NR.";
416 ZD  z1=z
417 48  GOSUB eingabe
418 4h  IF (z=0) OR (z=z1)THEN LINE (64*8,24*8)-(630,28*8),7,bf:RET
```

```
URN
PRINT z:COLOR 1,0
IF (st(1,z1)<2) OR (na(z)<>1) OR (na(z1)<>1) OR s2(1)
>50 THEN BEEP:RETURN
er=SQR(((kx(z)-kx(z1))/2)^2+(ky(z)-ky(z1))^2)
IF er > sr(f1(1)) THEN BEEP:RETURN
GET (126,89)-(444,180),gf&
WINDOW 2,"Transportschiffe senden :",(130,100)-(440,178),0,
1:GOSUB sf
n$=" Von Planet NR.":GOSUB text:PRINT z1
PRINT "  zu Planet NR.";z:PRINT
PRINT "  Stationierte Transportschiffe   :";tr(z1)
PRINT "  Transportable Rohstoffe        :";
IF st(1,z1)<3 THEN l1=0:ELSE l1=la(z1)
PRINT l1:IF st(1,z1)>2 THEN PRINT "";
PRINT "  Wieviele Transportschiffe  ";:GOSUB of:INPUT a1:GO
SUB sf
PRINT "  Wieviele Rohstoffe          ";:GOSUB of:INPUT a2:GO
SUB sf
IF a1>tr(z1) THEN a1=tr(z1)
IF a2>l1 THEN a2=l1
IF a2>a1 THEN a2=a1
IF a1<0 THEN a1=0
IF a2<0 THEN a2=0
de2(1,s2(1))=z:k2(1,s2(1))=a1:tr(z1)=tr(z1)-a1
k3(1,s2(1))=a2:la(z1)=la(z1)-a2:s2(1)=s2(1)+1
ens:
WINDOW CLOSE 2:PUT (126,89),gf&
RETURN
sf:
CALL setfont (WINDOW(8),font2&):RETURN
kh:
LINE (64*8,24*8)-(630,28*8),7,bf
COLOR 1,7:LOCATE 25,65:BEEP:CALL setfont (WINDOW(8),font2&)
PRINT "Kein Handel":LOCATE 26,65:PRINT "möglich":COLOR 1,0:
RETURN
hand:
GOSUB eingabe
IF z=0 THEN GOTO bau2:
IF na(z)<>1 THEN hand
GET (196,89)-(444,162),gf&
WINDOW 2,"Handelszentrum bauen",(200,100)-(440,160),0,1
PRINT:PRINT:GOSUB sf
IF st(1,z)>2 THEN n$=" Handelszentrum vorhanden":GOSUB tex
t:GOTO han3
IF v(1)<50 THEN n$=" Zu wenig geld":GOSUB text:GOTO han2
v(1)=v(1)-50:st(1,z)=3
han1:
n$= " Handelszentrum errichtet":GOSUB text
han2:
PRINT:PRINT "  Kosten : 50 ":
han3:
PRINT "  Vermögen :";v(1)
x=60:y=36:GOSUB ok:WINDOW CLOSE 2:PUT (196,89),gf&:GOTO han
d
indu:
GOSUB eingabe
IF z=0 THEN GOTO bau2:
IF na(z)<>1 THEN indu
WINDOW 2,"Industriezentrum bauen",(180,100)-(440,160),2,1
PRINT:PRINT:GOSUB sf
IF st(1,z)>3 THEN n$=" Industriezentrum vorhanden":GOSUB t
ext:GOTO ind3
IF v(1)<500 THEN n$=" Zu wenig geld":GOSUB text:GOTO ind2
v(1)=v(1)-500:st(1,heim(1))=3:st(1,z)=4:heim(1)=z
ind1:
n$=" Industriezentrum errichtet":GOSUB text
ind2:
PRINT:PRINT "  Kosten : 500":PRINT "  Vermögen : ";v(1)
ind3:
x=70:y=36:GOSUB ok:WINDOW CLOSE 2:GOSUB dis:GOTO indu
nex:
WINDOW 2,"",(200,100)-(440,170),2,1
PRINT:GOSUB sf:PRINT " Nächster Herrscher :"
PRINT:PRINT "    ";na$(1):x=60:y=50:GOSUB ok:WINDOW CLOSE 2
RETURN
erf:
IF jahr<>f4(1) THEN RETURN
f4(1)=jahr+INT(RND(1)*20)-f5(1)*3:IF f4(1)<5+jahr THEN f4(
1)=5+jahr
zu=RND(1):IF f1(1)=1 THEN zu=0
IF (zu<.5) AND (f1(1)<3) THEN f1(1)=f1(1)+1:RETURN
zu=RND(1)
IF zu<.4 THEN f3(1)=f3(1)+f5(1)-.9:RETURN
f2(1)=f2(1)+f5(1)-.9:RETURN
ok:
w=MOUSE(0):WINDOW OUTPUT 1:COLOR 1,4: LOCATE 2,65
PRINT "    weiter    ":GET (510,4)-(626,20),a%
LOCATE 2,65: PRINT "Bitte wählen  ":COLOR 1,0
WINDOW OUTPUT 2:PUT (x,y),a%:LINE (x+2,y+12)-(x+112,y+13),4
,bf
ok1:
w=MOUSE (0)
IF w<1 THEN ok1
RETURN
dis:
LINE (0,0)-(505,240),0,bf
IF di=0 THEN GOSUB puty:RETURN
FOR h= 1 TO ha(di) STEP 2:x=ve(h):y=ve(h+1)
  IF (st(1,x)>0) OR (st(1,y)>0) THEN LINE (kx(x),ky(x))-(
  kx(y),ky(y)),1:
NEXT h
GOSUB puty:CIRCLE (kx(heim(1))-1,ky(heim(1))-1),10,1,3,.05,
.3
RETURN
rev:
IF h=1 THEN re(pl)=0:RETURN
WINDOW 2,"Revolution",(200,80)-(440,160),2,1:
n$=" Auf Planet ":GOSUB text:PRINT
PRINT "  NR.";pl;" (";INT(kx(pl)/2);"/";INT(ky(pl));")"
PRINT "  fand eine Revolution statt.":PRINT
rs = INT((ro(pl)+mo(pl))/2+re(pl)/3)
IF rs<sh(pl) THEN PRINT "  Sie wurde niedergemetzelt.":sh(
pl)=INT(sh(pl)-rs/3*2):GOTO reve
sh(pl)=INT(rs-sh(pl)/3*2):tr(pl)=0:st(na(pl),pl)=1:na(pl)=0
PRINT "  Sie war erfolgreich."
reve:
re(pl)=0:x=60:y=58:GOSUB ok
WINDOW CLOSE 2:RETURN
revo:
IF st(1,z)<1 THEN RETURN:ELSE:GET (196,69)-(442,182),gf&
WINDOW 2,"Revolution unterstützen",(200,80)-(440,180),0,1
n$=" Auf Planet NR.":GOSUB text:PRINT z
PRINT "  Vermögen  : ";v(1):PRINT:PRINT "  Wieviel ihres Ge
ldes"
PRINT "  wollen Sie investieren ";:GOSUB of:INPUT g:GOSUB s
f
IF g>v(1) THEN g=v(1)
IF g<0 THEN g=0
PRINT:PRINT "  O.K. Vermögen :";v(1)-g
v(1)=v(1)-g:re(z)=re(z)+g
x=60:y=75:GOSUB ok:WINDOW CLOSE 2:PUT (196,69),gf&:RETURN
beenden:
WINDOW 2,"Spiel Beenden !!!!!  ",(200,80)-(440,140),2,1
n$= " Sind sie sicher ?":GOSUB text:PRINT
PRINT "  (j/n) ";:GOSUB of:INPUT a$
IF a$="j" OR a$="J" THEN WINDOW CLOSE 2:GOTO start
WINDOW CLOSE 2:GOTO ww6
sav:
WINDOW 2,"Spielstand speichern",(200,80)-(440,140),2,1
n$=" Bitte geben sie einen":GOSUB text:PRINT
PRINT "  Namen ein ";:GOSUB of:INPUT s$:IF s$="" THEN savn
OPEN s$ FOR OUTPUT AS #3
PRINT#3,jahr,spieler,f$
FOR p= 1 TO 59:
PRINT#3,mo(p),sh(p),tr(p),ro(p),na(p),re(p),la(p)
NEXT
FOR y = 1 TO spieler
  FOR x= 1 TO 59:PRINT#3,st(y,x)
NEXT x,y
FOR x= 1 TO spieler
  PRINT#3,na$(x)
  PRINT#3,heim(x),f1(x),f2(x),f3(x),f4(x),f5(x),v(x)
NEXT x
CLOSE 3
savn:
WINDOW CLOSE 2
RETURN
lod:
WINDOW 2,"Spielstand laden",(200,80)-(440,140),2,1
n$=" Bitte geben sie den":GOSUB text
PRINT:PRINT "  Namen des Spiels ein":GOSUB of
INPUT "  ";s$:IF s$ ="" THEN lode
```

Listing. »Fantasy« versetzt Sie in das Jahr 6533. Erobern Sie neue Welten (Fortsetzung).

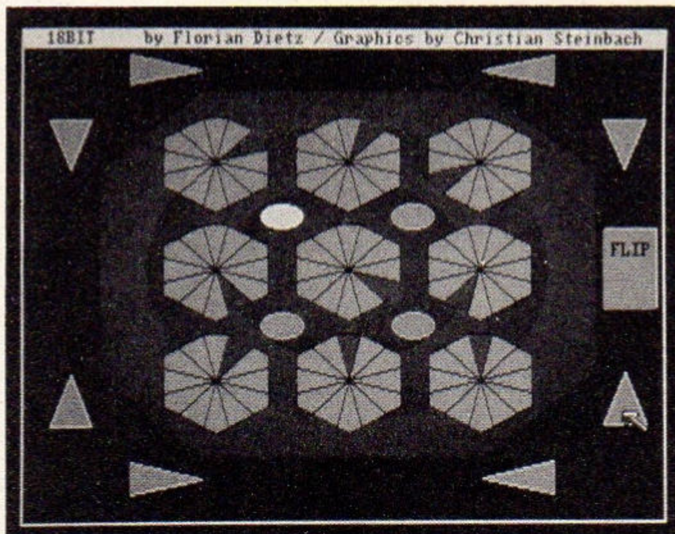
```
Mc  OPEN s$ FOR INPUT AS #3
ma  INPUT#3,jahr,spieler,f$
aU  FOR p= 1 TO 59
062   INPUT#3,mo(p),sh(p),tr(p),ro(p),na(p),re(p),la(p)
rE0 NEXT p
o6  FOR y = 1 TO spieler
oG2   FOR x= 1 TO 59:INPUT#3,st(y,x)
UcO NEXT x,y
vG  FOR x= 1 TO spieler
WK2   INPUT#3,na$(x),heim(x),f1(x),f2(x),f3(x),f4(x),f5(x),v(x
      )
D10 NEXT x
Dz  CLOSE 3
VP  GOSUB wwlod
57  lode:
YE  WINDOW CLOSE 2
JL  RETURN
Oy  graf:
sN  GOSUB menue8
Oy  graf1:
mq  GOSUB eingabe
OP  IF m<1 OR m>5 THEN GOSUB menue2:RETURN
Mk  ON m GOSUB dars,p1,p2,p3,puty:CIRCLE (kx(heim(1))-1,ky(heim
    (1))-1),10,1,3,.05,.3
uR  GOTO graf1
FY  pro:
4I  LINE (0,0)-(505,240),0,bf
NV  IF di=0 THEN GOSUB puty:RETURN
ZW  FOR h= 1 TO ha(di) STEP 2:x=ve(h):y=ve(h+1)
rE2   IF (st(1,x)>0) OR (st(1,y)>0) THEN LINE (kx(x),ky(x))-(
      kx(y),ky(y)),1:
zE0 NEXT h
en  ON m GOSUB p1,p1,p2,p3,puty:CIRCLE (kx(heim(1))-1,ky(heim(1
    ))-1),10,1,3,.05,.3
ya  RETURN
ih  p1:
Ii  FOR h= 1 TO 59
o12   IF POINT(kx(h),ky(h))=0 THEN pt1
2G    IF st(1,h)<2 THEN PUT (kx(h)-8,ky(h)-4),a4,PSET:GOTO pt1
H9    IF la(h)>150 THEN PUT (kx(h)-8,ky(h)-4),a3,PSET:GOTO pt1
hl    IF la(h)>30 THEN PUT (kx(h)-8,ky(h)-4),a2,PSET:GOTO pt1
MB    PUT (kx(h)-8,ky(h)-4),a1,PSET
av0 pt1:
AP  NEXT h
8k  RETURN
vv  p2:
Ss  FOR h= 1 TO 59
1z2   IF POINT(kx(h),ky(h))=0 THEN pt2
ET    IF st(1,h)<2 THEN PUT (kx(h)-8,ky(h)-4),a4,PSET:GOTO pt2
SN    IF tr(h)>50 THEN PUT (kx(h)-8,ky(h)-4),a3,PSET:GOTO pt2
3D    IF tr(h)>10 THEN PUT (kx(h)-8,ky(h)-4),a2,PSET:GOTO pt2
WL    PUT (kx(h)-8,ky(h)-4),a1,PSET
oA0 pt2:
KZ  NEXT h
Iu  RETURN
89  p3:
c2  FOR h= 1 TO 59
ED2   IF POINT(kx(h),ky(h))=0 THEN pt3
Qg    IF st(1,h)<2 THEN PUT (kx(h)-8,ky(h)-4),a4,PSET:GOTO pt3
f9    IF sh(h)>50 THEN PUT (kx(h)-8,ky(h)-4),a3,PSET:GOTO pt3
Gz    IF sh(h)>10 THEN PUT (kx(h)-8,ky(h)-4),a2,PSET:GOTO pt3
gV    PUT (kx(h)-8,ky(h)-4),a1,PSET
2P0 pt3:
UJ  NEXT h
S4  RETURN
Wr  font:
TN  name0$="sapphire.font"+CHR$(0)
Zs  h% = 14
li  textattr&(0) = SADD(name0$):textattr&(1) = h%*2^16+0*2^4+0
6e  font1& = openfont&(VARPTR(textattr&(0)))
Ov  IF font1&=0 THEN font1& = opendiskfont&(VARPTR(textattr&(0)
    ))
zg  name0$="opal.font"+CHR$(0):h% = 9
pm  textattr&(0) = SADD(name0$):textattr&(1) = h%*2^16+0*2^4+0
Gk  font2& = openfont&(VARPTR(textattr&(0)))
Z8  IF font2&=0 THEN font2& = opendiskfont&(VARPTR(textattr&(0)
    ))
90  name0$="topaz.font"+CHR$(0):h% = 8
tq  textattr&(0) = SADD(name0$):textattr&(1) = h%*2^16+0*2^4+0
8k  font0& = openfont&(VARPTR(textattr&(0)))
OF  name0$="diamond.font"+CHR$(0):h% = 20
wt  textattr&(0) = SADD(name0$):textattr&(1) = h%*2^16+0*2^4+0
Tt  font3& = openfont&(VARPTR(textattr&(0)))
n0  IF font3&=0 THEN font3& = opendiskfont&(VARPTR(textattr&(0)
    ))
kM  RETURN
1q  of:
2t  CALL setfont (WINDOW(8),font0&):RETURN
S1  text:
Z1  CALL setfont (WINDOW(8),font1&):CALL move(WINDOW(8),1,13)
X0  COLOR 7,0:PRINT LEFT$(n$,2);:CALL setfont (WINDOW(8),font2&
    )
Kw  COLOR 1,0:PRINT RIGHT$(n$,LEN(n$)-2);
rT  RETURN
QA  cfont:
Ak  CALL closefont (font1&):CALL closefont (font2&)
3n  CALL closefont (font3&):CALL closefont (font0&)
vX  RETURN
Ua  sieg:
UI  GOSUB titel1
tI  FOR x=1 TO spieler:IF na(heim(x))=x THEN wi=x
Y3  NEXT x
X0  PALETTE 4,0,0,1:PALETTE 5,0,0,.5:x%=90:y%=120:COLOR 6,0
1K  CALL setfont (WINDOW(8),font2&):POKEW WINDOW(8)+64,1
D3  ns%=SetSoftStyle&(WINDOW(8),2,en%)
SM  CALL move(WINDOW(8),x%,y%):PRINT "Gewonnen hat :":COLOR 7,0
Vg  CALL move(WINDOW(8),x%+1,y%+1):PRINT "Gewonnen hat :"
yS  x%=70:y%=150:COLOR 4,0:CALL setfont (WINDOW(8),font3&)
TB  POKEW WINDOW(8)+64,2:ns%=SetSoftStyle&(WINDOW(8),4,en%)
c7  CALL move(WINDOW(8),x%,y%):PRINT na$(wi):COLOR 5,0
dG  CALL move(WINDOW(8),x%+1,y%+1):PRINT na$(wi)
9e  ttu:
1M  w=MOUSE (0)
KX  IF w<1 THEN ttu
Y1  CALL SetDrMd(WINDOW(8),1)
AD  WINDOW CLOSE 6 :SCREEN CLOSE 2
my  GOSUB titel:GOSUB ssr:GOTO neus
Jp  titel:
MP  GOSUB titel1:GOTO titel2
L9  titel2:
uI  x%=60:y%=120:COLOR 4,0:CALL setfont (WINDOW(8),font1&)
U9  POKEW WINDOW(8)+64,1:CALL move(WINDOW(8),x%,y%)
3G  PRINT "Written in 1988 by:":COLOR 5,0
6Z  CALL move(WINDOW(8),x%+1,y%+1):PRINT "Written in 1988 by:"
Je  x%=50:y%=160:COLOR 6,0:CALL setfont (WINDOW(8),font3&)
kS  POKEW WINDOW(8)+64,2:ns%=SetSoftStyle&(WINDOW(8),4,en%)
Ut  CALL move(WINDOW(8),x%,y%):PRINT "Holger Hoffmann":COLOR 7,
    0
50  CALL move(WINDOW(8),x%+1,y%+1):PRINT "Holger Hoffmann"
ex  tti:
2d  w=MOUSE (0)
RQ  IF w<1 THEN tti
pz  CALL SetDrMd(WINDOW(8),1)
F5  WINDOW CLOSE 6 :SCREEN CLOSE 2:RETURN
XK  titel1:
a7  SCREEN 2,320,256,3,1:WINDOW 6,"",(0,0)-(310,240),0,2
8D  CALL setfont (WINDOW(8),font3&):en%=AskSoftStyle&(WINDOW(8)
    )
Hf  ns%=SetSoftStyle&(WINDOW(8),7,en%):CALL SetDrMd(WINDOW(8),0
    )
Lh  POKEW WINDOW(8)+64,4:PALETTE 0,0,0,0:PALETTE 1,0,0,0
6Y  PALETTE 2,1,0,1:PALETTE 3,.5,0,.5:PALETTE 4,0,1,0
S5  PALETTE 5,0,.5,0:PALETTE 6,1,1,0:PALETTE 7,1,.5,0
Ab  x%=62:y%=65:COLOR 2,0:CALL move(WINDOW(8),x%,y%)
0g  PRINT "  FANTASY  ":COLOR 3,0:CALL move(WINDOW(8),x%+1,y%+1
    )
9H  PRINT "  FANTASY  ":LINE (0,73)-(318,73),2
as  LINE (1,74)-(319,74),3
gI  RETURN
Qq  ssr:
12  SCREEN  1,640,256,3,2
gl  WINDOW 1,"Fantasy  Written in 1988 by Holger Hoffmann",,0,1
QZ  MENU 1,0,1,"":MENU 2,0,1,"":MENU 3,0,1,"":MENU 4,0,1,""
1N  RETURN
NI  fehler:
qm  WINDOW 2,"Fehlermeldung",(200,100)-(360,175),2,1
WB  n$=" Error !!!!":GOSUB text
NE  PRINT:PRINT:IF ERR<>53 THEN PRINT "   Fehler Nr.";ERR:ELS
    E PRINT "   File not found"
21  PRINT:PRINT "   Mit Return":PRINT "   bestätigen ";:COLOR 0
    ,0:INPUT n$
ro  WINDOW CLOSE 2:RESUME start
KG  REM Ende von FANTASY
ry  REM Written in 1988 by Holger Hoffmann
(C) 1989 M&T
```

Listing. »Fantasy« versetzt Sie in das Jahr 6533. Erobern Sie neue Welten (Schluß).

Trainingslager...

Muskeln stählt man im Bodybuilding-Center mit Hanteln. Die »Hanteln«, um das Gehirn zu trainieren, sind Denkspiele. Folgen Sie uns jetzt in den Übungsraum.

Unsere Übung heißt »18-BIT« und fördert das räumliche Vorstellungsvermögen. Im Bild sehen Sie das Trainingsgerät. Es handelt sich um neun Zeiger und vier Knöpfe. Die Aufgabe ist es, alle Zeiger so zu drehen, daß sie nach oben zeigen. Sie glauben, das sei einfach? Weit gefehlt, denn die Zeiger sind untereinander verbunden. Außerdem befinden sich auf der »Rückseite« des achteckigen Feldes noch einmal neun Zeiger. Natürlich müssen auch diese nach oben zeigen. Es sind also insgesamt 18 Zeiger, woher auch der Name des Spiels stammt. Die zwei Seiten sind farblich (blau und grün) unterschieden. Die Rückseite sehen Sie, wenn Sie den Schalter mit der Aufschrift »FLIP« anklicken.



Die eine Seite von »18BIT« mit neun Zeigern und vier Schaltern. Finden Sie eine Lösung?

Um den angestrebten Zustand zu erreichen, stehen Ihnen an den vier Eckpunkten acht Schalter zur Verfügung. Mit ihnen lassen sich die Anzeigen verstellen. Je nach Pfeilrichtung im oder gegen den Uhrzeigersinn. Jetzt kommt das räumliche Vorstellungsvermögen zum Zug. Die Anzeigen auf der Vorder- und Rückseite sind ebenfalls miteinander verbunden. Dreht man die rechte untere Anzeige nach links, bewegt sich der entsprechende Zeiger (Rückseite links unten) entgegengesetzt. Welche Anzeigen sich automatisch mitdrehen, läßt sich über die vier Knöpfe zwischen den Anzeigen beeinflussen. Gelbe Knöpfe sind aktiv, lila Knöpfe sind inaktiv. Auf der anderen Seite ist der Zustand der Knöpfe genau umgekehrt. Wieder sind die Positionen seitenverkehrt.

Die Einstellung der Knöpfe ist entscheidend für das Verhalten der Anzeigen. Sind alle Knöpfe aktiv, drehen sich alle Zeiger mit, egal an welcher Ecke gedreht wird. Stellen Sie alle Knöpfe ab, bewegen sich die vier Scheiben an den Ecken des Spielfeldes. Ein Knopf stellt also immer die Verbindung zwischen den vier ihn umgebenden Anzeigen her. Wenn Sie jetzt den Eindruck haben, daß 18BIT schwierig zu lösen ist, haben Sie recht. Möglich ist es aber auf jeden Fall. Nach längerem Spiel bekommt man langsam ein Gefühl für das Problem.

Außer der Bedienung mit den Schaltern besitzt 18BIT noch drei Pull-Down-Menüs für andere Funktionen, die wir Ihnen in den folgenden Punkten ausführlich vorstellen wollen:

■ Project/New
Mit diesem Menüpunkt starten Sie ein neues Spiel. Die Zeiger werden zufällig auf neue Positionen gesetzt.

■ Project/Save
Dient zum Speichern des Spielstands unter einem vorher (mit Save As) angegebenen Namen. Ist noch kein Name eingegeben worden, fragt 18BIT automatisch danach.

■ Project/Save As
Während des Spiels kann gespeichert werden, wobei ein neuer Name angegeben wird. Dadurch können mehrere Spielstände bei einem Spiel gesichert werden.

■ Project/Quicksave
Um schnell einen Zustand zu speichern, verwenden Sie diesen Menüpunkt. Der Name der Datei lautet »18BIT.quick«. Der alte mit »Quicksave« gespeicherte Spielstand wird überschrieben.

■ Project/Load
Gespeicherte Spielstände können geladen werden. Vor dem Laden fragt 18BIT nach dem Dateinamen. Wie bei allen anderen Dateifunktionen ist es möglich, einen ganzen Pfad anzugeben.

■ Project/Quickload
Lädt den Spielstand in der Datei »18BIT.quick«.

■ Project/Quit
Beendet das Spiel, verläßt aber nicht den Amiga-Basic-Interpreter.

18BIT bietet Ihnen eine Hilfe, die Sie sicher bald nicht mehr missen wollen: Sequenzen. Wenn Sie für eine Problemstellung einen Lösungsweg gefunden haben, können Sie diese Sequenz auf Diskette speichern und später wieder abrufen.

■ Edit/SeqStart
Hiermit schalten Sie 18BIT auf »Aufnahme«. Das Programm merkt sich von nun ab alle Drehungen, Seitenwechsel und Knopfeinstellungen. Vor der Eingabe der Spielzüge verlangt 18BIT noch einen Namen für die anzulegende Datei.

■ Edit/SeqStop
Beendet den Aufnahmemodus und speichert die Daten in der Datei mit dem angegebenen Namen.

■ Edit/SeqShow
Fragt nach dem Namen der Datei und lädt die Sequenz, die dann automatisch durchgeführt wird.

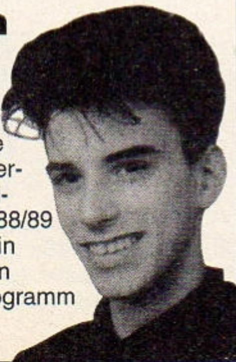
■ System/Sound On
Schaltet den Ton bei Drehung der Anzeigen an. Durch das akustische Signal ist es leichter, zu verfolgen, wieviele Anzeigen verstellt werden.

■ System/Sound Off
Der Sound wird ausgeschaltet, was bewirkt, daß das Spiel etwas schneller läuft.

■ System/Cycling On
18BIT verändert die Farben der Kreise, wodurch der Eindruck von Bewegung entsteht. Beim Verstellen von Zeigern

Florian Dietz/Christian Steinbach





Beide Autoren von 18BIT besuchen die 13. Klasse des Uhland-Gymnasiums in Tübingen. Die Freundschaft zwischen beiden begann im Jahr 1986, in dem sie auch je einen C 64 bekamen. Nach Basic folgte die Programmierung in Comal. Vor ca. eineinhalb Jahren erfolgte dann der Aufstieg zum Amiga. Nach der Einarbeitung in Amiga-Basic entstand in den Weihnachtsferien 88/89 das Programm des Monats. Jetzt lernen die beiden C. Ein weiteres Projekt ist schon in Vorbereitung. Die Redaktion wünscht beiden, daß dies ein ebenso erfolgreiches Programm wird wie das letzte.

oder bei einem Seitenwechsel wird das Colorcycling (Farbrotation) kurz angehalten.

■ System/Cycling Off

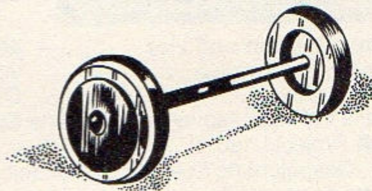
Ohne sich verändernde Farben läuft 18BIT nicht schneller. Für die Augen ist diese Einstellung bei längerem Spiel angenehmer.

18BIT läßt sich mit dem AC-Basic-Compiler übersetzen. Die Schalter »N« und »R« müssen dabei eingeschaltet sein.

Jetzt macht »Gehirntraining« wirklich Spaß. Passen Sie auf, daß Sie keinen Muskelkater bekommen. *René Beaupoil/kn*

Programmname:	18BIT
Computer:	A500, A1000, A2000 m. Kickstart 1.2 u. 1.3
Sprache:	Amiga-Basic 1.2
Bemerkung:	siehe Text

```
Programmautor: Florian Dietz/Christian Steinbach
------------------------------------------------
  1 X30 REM ** 18BIT Programmed by Florian Dietz
  2 IO  REM ** Graphics by Christian Steinbach
  3 wT  CLEAR ,27000,5000
  4 iU  GOSUB HauptProgramm
  5 pk  END
  6 OE  REM Hauptprogramm:
  7 mt  HauptProgramm:
  8 sm2   GOSUB VariableDefining
  9 HA    GOSUB OutputDefinition
 10 rt    GOSUB MenuDefining
 11 6f    ON MENU GOSUB MenuTrapping
 12 Ta    ON MOUSE GOSUB MouseTrapping
 13 mO    ON BREAK GOSUB BreakTrapping
 14 KQ    MOUSE STOP
 15 nu    MENU STOP
 16 FH    BREAK STOP
 17 08    GOSUB Graphic
 18 OU    MOUSE STOP
 19 ry    MENU STOP
 20 bx    GOSUB PositionPaint
 21 fG    Cycle=1
 22 pO    GOSUB Blue
 23 Sj    MOUSE ON
 24 c2    MENU ON
 25 Na    BREAK ON
 26 dd    WHILE -1
 27 sg    WEND
 28 oQ0 RETURN
 29 Hx  REM Belegung und Dimensionierung von Variablen und Feldern:
 30 UW  VariableDefining:
 31 8E2   TIMER ON
 32 8X    OPTION BASE 1
 33 dU    DIM Display(3,3,2), OldDisplay(3,3,2)
 34 Os    DIM Button$(4), Button(4,2)
 35 1C    DIM Flipped(3,3,2)
 36 OW    DIM xpos(12),ypos(12)
 37 mT    DIM xpos2(6),ypos2(6)
 38 1z    RESTORE SectorData
 39 My    FOR x=1 TO 12
 40 NJ4     READ xpos(x),ypos(x)
 41 a52   NEXT x
 42 Xn    RESTORE SectorData2
 43 Vu    FOR x=1 TO 6
 44 KO4     READ xpos2(x),ypos2(x)
 45 e92   NEXT x
 46 De    Button$(1)="122111"
 47 Wx    Button$(2)="213231"
 48 cE    Button$(3)="231213"
 49 vX    Button$(4)="322333"
 50 Wt    FOR x=1 TO 4
 51 Xo4     FOR y=1 TO 2
 52 B86       Button(x,y)=y-1
 53 oK4     NEXT y
 54 nI2   NEXT x
 55 4D    RANDOMIZE TIMER
 56 Zv    FOR x=1 TO 3
 57 gy4     FOR y=1 TO 3
 58 kx6       FOR z=1 TO 2
 59 1H8         Display(x,y,z)=INT(1+12*RND)
 60 uW          OldDisplay(x,y,z)=Display(x,y,z)
 61 yV6       NEXT z
 62 xT4     NEXT y
 63 wR2   NEXT x
```



```
 64 oI    FOR x=1 TO 3 STEP 2
 65 vL4     FOR y=1 TO 3 STEP 2
 66 Ox6       Display(x,y,2)=12-Display(x,y,1)
 67 dp        IF Display(x,y,2)=0 THEN Display(x,y,2)=12
 68 3Z4     NEXT y
 69 2X2   NEXT x
 70 JS    RANDOMIZE TIMER
 71 Vo    DIM Welle%(256)
 72 mm    FOR x=0 TO 255
 73 L84     Welle%(x+1)=RND*255-128
 74 7c2   NEXT x
 75 ja    WAVE 0,Welle%
 76 jc    ERASE Welle%
 77 nS    DIM Fa(11)
 78 g1    Farbe=1
 79 Eg    FOR x=1 TO 9
 80 dY4     Fa(x)=Farbe-1/16
 81 SL      Farbe=Farbe-1/16
 82 Fk2   NEXT x
 83 gR    Filename$=""
 84 C8    Oldname$=""
 85 Fb    Sound1=1
 86 kM0 RETURN
 87 R3  REM Oeffnen eines neuen Screens und Neudefinition des Ausga
        bewindows:
 88 Xq  OutputDefinition:
 89 1Z2   SCREEN 1,520,240,4,2
 90 8w    WINDOW 1," 18BIT .  by Florian Dietz / Graphics by Chri
          stian Steinbach",,16,1
 91 pRO RETURN
 92 j7  REM Belegung der Menues und Start des Event Trappings fuer
        die Menues:
 93 yu  MenuDefining:
 94 dD2   MENU 1,0,1,"Project"
 95 Oe    MENU 1,1,1,"New      "
 96 b4    MENU 1,2,1,"Save     "
 97 Vr    MENU 1,3,1,"Save As  "
 98 hS    MENU 1,4,1,"Quicksave"
 99 SQ    MENU 1,5,1,"Load     "
100 Ti    MENU 1,6,1,"Quickload"
101 id    MENU 1,7,1,"Quit     "
102 AW    MENU 2,0,1,"Edit"
103 um    MENU 2,1,1,"SeqStart"
104 b6    MENU 2,2,1,"SeqStop "
105 f6    MENU 2,3,1,"SeqShow "
106 Cq    MENU 3,0,1,"System"
107 As    MENU 3,1,2,"  Sound On   "
108 H4    MENU 3,2,1,"  Sound Off  "
109 NB    MENU 3,3,2,"  Cycling On "
110 Uj    MENU 3,4,1,"  Cycling Off"
111 fm    MENU 4,0,0,""
112 AmO RETURN
113 7i  REM Verzweigung zu den einzelnen Menuepunkten:
114 ep  MenuTrapping:
115 x32   MOUSE STOP
116 QX    MENU STOP
117 jp    Men=MENU(0)
118 md    Punkt=MENU(1)
119 Du    IF Men=1 THEN ON Punkt GOSUB New1,Save1,SaveAs1,Quicksave
          1,Load1,Quickload1,Quit1
120 iw    IF Men=2 THEN ON Punkt GOSUB SeqStart1,SeqStop1,SeqShow1
121 Kb    IF Men=3 THEN ON Punkt GOSUB SoundOn1,SoundOff1,CyclingOn
          1,CyclingOff1
122 3K    MOUSE ON
123 Dd    MENU ON
124 MyO RETURN
125 nQ  REM Auswertung der Maussteuerung:
126 Ob  MouseTrapping:
127 Zx2   LTaste=MOUSE(0)
128 KW    x=MOUSE(1)
129 Sg    y=MOUSE(2)
130 CI    MOUSE STOP
131 fm    MENU STOP
132 nf    IF x>466 AND x<511 AND y>88 AND y<129 THEN GOSUB Scre
          enChange
133 Ub    IF LTaste<>0 AND WINDOW(0)=1 AND Seite=0 THEN
134 Z54     IF x>84 AND x<145 AND y>1 AND y<18 THEN Richtung=1
            : Hebel=1 : GOSUB DisplayPosition
135 DU      IF x>18 AND x<57 AND y>34 AND y<61 THEN Richtung=-1
             : Hebel=1 : GOSUB DisplayPosition
136 8X      IF x>366 AND x<427 AND y>1 AND y<18 THEN Richtung=-
            1 : Hebel=2 : GOSUB DisplayPosition
```

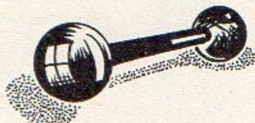


```
137 xT   IF x>464 AND x<503 AND y>34 AND y<61 THEN Richtung=1 : Hebel=2 : GOSUB DisplayPosition
138 fY   IF x>84 AND x<145 AND y>199 AND y<216 THEN Richtung=-1 : Hebel=3 : GOSUB DisplayPosition
139 Hd   IF x>18 AND x<57 AND y>157 AND y<184 THEN Richtung=1 : Hebel=3 : GOSUB DisplayPosition
140 WI   IF x>366 AND x<427 AND y>199 AND y<216 THEN Richtung=1 : Hebel=4 : GOSUB DisplayPosition
141 9u   IF x>464 AND x<503 AND y>157 AND y<184 THEN Richtung=-1 : Hebel=4 : GOSUB DisplayPosition
142 Zx   IF x>190 AND x<227 AND y>76 AND y<90 THEN
143 yu6    Button(1,1)=1-Button(1,1) : Button(1,2)=1-Button(1,1) : PAINT (207,83),2+Button(1,1),13
144 O94  END IF
145 b0   IF x>294 AND x<331 AND y>76 AND y<90 THEN
146 J76    Button(2,1)=1-Button(2,1) : Button(2,2)=1-Button(2,1) : PAINT (312,83),2+Button(2,1),13
147 JC4  END IF
148 Ad   IF x>190 AND x<227 AND y>129 AND y<143 THEN
149 tJ6    Button(3,1)=1-Button(3,1) : Button(3,2)=1-Button(3,1) : PAINT (207,136),2+Button(3,1),13
150 MF4  END IF
151 Cg   IF x>294 AND x<331 AND y>129 AND y<143 THEN
152 Fx6    Button(4,1)=1-Button(4,1) : Button(4,2)=1-Button(4,1) : PAINT (312,136),2+Button(4,1),13
153 PI4  END IF
154 6F2 ELSEIF LTaste<>0 AND WINDOW(0)=1 AND Seite=1 THEN
155 QQ4  IF x>84 AND x<145 AND y>1 AND y<18 THEN Richtung=-1 : Hebel=2 : GOSUB DisplayPosition
156 Au   IF x>18 AND x<57 AND y>34 AND y<61 THEN Richtung=1 : Hebel=2 : GOSUB DisplayPosition
157 by   IF x>366 AND x<427 AND y>2 AND y<19 THEN Richtung=1 : Hebel=1 : GOSUB DisplayPosition
158 AS   IF x>464 AND x<503 AND y>34 AND y<61 THEN Richtung=-1 : Hebel=1 : GOSUB DisplayPosition
159 Ko   IF x>84 AND x<145 AND y>199 AND y<216 THEN Richtung=1 : Hebel=4 : GOSUB DisplayPosition
160 dn   IF x>18 AND x<57 AND y>157 AND y<184 THEN Richtung=-1 : Hebel=4 : GOSUB DisplayPosition
161 1i   IF x>366 AND x<427 AND y>200 AND y<217 THEN Richtung=-1 : Hebel=3 : GOSUB DisplayPosition
162 Er   IF x>464 AND x<503 AND y>157 AND y<184 THEN Richtung=1 : Hebel=3 : GOSUB DisplayPosition
163 uI   IF x>190 AND x<227 AND y>76 AND y<90 THEN
164 eb6    Button(2,1)=1-Button(2,1) : Button(2,2)=1-Button(2,1) : PAINT (207,83),2+Button(2,2),13
165 bU4  END IF
166 wL   IF x>294 AND x<331 AND y>76 AND y<90 THEN
167 RG6    Button(1,1)=1-Button(1,1) : Button(1,2)=1-Button(1,1) : PAINT (312,83),2+Button(1,2),13
168 eX4  END IF
169 Vy   IF x>190 AND x<227 AND y>129 AND y<143 THEN
170 bS6    Button(4,1)=1-Button(4,1) : Button(4,2)=1-Button(4,1) : PAINT (207,136),2+Button(4,2),13
171 ha4  END IF
172 X1   IF x>294 AND x<331 AND y>129 AND y<143 THEN
173 N66    Button(3,1)=1-Button(3,1) : Button(3,2)=1-Button(3,1) : PAINT (312,136),2+Button(3,2),13
174 kd4  END IF
175 1e2 END IF
176 vC  MOUSE ON
177 5V  MENU ON
178 Eq0 RETURN
179 hM  REM Farbfestlegung fuer die Grundfarben beider Seiten:
180 Eg  ColourDefining:
181 Rf2  RESTORE ColourData
182 MU   FOR x = 0 TO 3
183 Yf4   READ r,g,b
184 rn    PALETTE x,r,g,b
185 uP2  NEXT x
186 Ja   FOR x = 13 TO 15
187 cJ4   READ r,g,b
188 vr    PALETTE x,r,g,b
189 yT2  NEXT x
190 Q20 RETURN
191 Ag  REM R-,G- und B-Werte fuer die Grundfarben:
192 2Q  ColourData:
193 Oe2  DATA  0,0,0,1,.8,.7,1,0,.8,1,1,0
194 Sc   DATA  .5,.5,.5,.7,.7,.7,1,0,.13
195 EU0 REM Farbfestlegung fuer die blaue/erste Seite:
196 V5  Blue:
197 UY2  IF Cycle=0 THEN
198 V74   FOR x=12 TO 4 STEP -1
199 3Z6    PALETTE x,0,0,Fa(x-3)
200 9e4   NEXT x
201 B42  END IF
202 fD   WHILE Cycle=1
203 ZC4   GOSUB Cycling
204 bD    FOR x=12 TO 4 STEP -1
205 9f6    PALETTE x,0,0,Fa(x-3)
206 Fk4   NEXT x
207 Qh    MOUSE ON
208 a0    MENU ON
209 TZ    MOUSE STOP
210 w3    MENU STOP
211 qe2  WEND
212 m00 RETURN
213 uo  REM Farbfestlegung fuer die gruene/zweite Seite:
214 5e  Green:
215 mq2  IF Cycle=0 THEN
216 nP4   FOR x=12 TO 4 STEP -1
217 o36    PALETTE x,0,Fa(x-3)-1/8,0
218 Rw4   NEXT x
219 TM2  END IF
220 xV   WHILE Cycle=1
221 rU4   GOSUB Cycling
222 tV    FOR x=12 TO 4 STEP -1
223 u96    PALETTE x,0,Fa(x-3)-1/8,0
224 X24   NEXT x
225 1z    MOUSE ON
226 sI    MENU ON
227 1r    MOUSE STOP
228 EL    MENU STOP
229 8w2  WEND
230 4g0 RETURN
231 e1  REM Colour Cycling-Routine fuer beide Seiten:
232 Kk  Cycling:
233 1v2  Fa(10)=Fa(1)
234 JB   FOR x=1 TO 9
235 XL4   Fa(x)=Fa(x+1)
236 ty2  NEXT
237 Bn0 RETURN
238 Q8  REM Redefinierung der Ausgaben:
239 WX  Quit1:
240 3d2  TIMER OFF
241 WH   MOUSE OFF
242 C6   MENU OFF
243 7V   MENU RESET
244 gb   END
245 Jv0 RETURN
246 HB  REM Variablen- und Screenwechsel auf die andere Seite:
247 Mv  ScreenChange:
248 F02  Seite=1-Seite
249 V1   GOSUB PositionChange
250 000 RETURN
251 Js  REM Berechnung der Displaystellungen:
252 ik  DisplayPosition:
253 6V2  IF Seq=1 THEN GOSUB Sequence
254 17   FOR x=1 TO 3
255 sA4   FOR y=1 TO 3
256 ir6    OldDisplay(x,y,Seite+1)=Display(x,y,Seite+1)
257 6c4   NEXT y
258 5a2  NEXT x
259 qv   AB=Button(Hebel,2)+1
260 YY   IF AB=2 THEN Richtung=-Richtung
261 vI   FOR x=1 TO 4
262 hD4   IF Button(x,AB)=1 THEN
263 OI6    FOR y=1 TO 3
264 wF8     x$=MID$(Button$(x),2*y-1,1)
265 eJ      y$=MID$(Button$(x),2*y,1)
266 UH      xx=VAL(x$)
267 eU      yy=VAL(y$)
268 bw      IF Flipped(xx,yy,AB)=0 THEN
269 PqA      Display(xx,yy,AB)=Display(xx,yy,AB)+Richtung
270 75       IF Display(xx,yy,AB)>12 THEN Display(xx,yy,AB)=1
271 9w       IF Display(xx,yy,AB)<1 THEN Display(xx,yy,AB)=12
272 D0       Flipped(xx,yy,AB)=1
273 LE8     END IF
274 Nt6    NEXT y
```



Listing. »Gehirntraining« mit »18BIT« ist eine spannende Angelegenheit. Bitte mit dem Checksummer (Seite 143) eingeben (Fortsetzung).

```
DD       x$=MID$(Button$(x),5,1)
MJ       y$=MID$(Button$(x),6,1)
fS       xx=VAL(x$)
pf       yy=VAL(y$)
Cg       Display(xx,yy,3-AB)=Display(xx,yy,3-AB)-Richtung
06       IF Display(xx,yy,3-AB)>12 THEN Display(xx,yy,3-AB)=1
sr       IF Display(xx,yy,3-AB)<1 THEN Display(xx,yy,3-AB)=12
UN4     END IF
Uz2   NEXT x
r6    Display(2,2,AB)=Display(2,2,AB)+Richtung
2N    IF Display(2,2,AB)>12 THEN Display(2,2,AB)=1
84    IF Display(2,2,AB)<1 THEN Display(2,2,AB)=12
Ks    GOSUB PositionOutput
Jf    FOR x=1 TO 3
Q14     FOR y=1 TO 3
gf6       Flipped(x,y,AB)=0
eA4     NEXT y
d82   NEXT x
5h0 RETURN
49  REM Routine zur Ausgabe des Screenwechsels von 18BIT:
2h  PositionChange:
1r2   IF Seite=0 THEN
nu4     IF POINT (207,83)<>2+Button(1,1) THEN PAINT (207,83),
        2+Button(1,1),13
TW      IF POINT (312,83)<>2+Button(2,1) THEN PAINT (312,83),
        2+Button(2,1),13
KR      IF POINT (207,136)<>2+Button(3,1) THEN PAINT (207,136
        ),2+Button(3,1),13
0t      IF POINT (312,136)<>2+Button(4,1) THEN PAINT (312,136
        ),2+Button(4,1),13
Ws      FOR x=1 TO 3
M06       x2=x-1
ew        FOR y=1 TO 3
3h8         IF Display(x,y,1)<>Display(4-x,y,2) THEN
zpA           Display=Display(4-x,y,2)
08            Col=13
X0            GOSUB Pointer
zH            Display=Display(x,y,1)
VH            Col=15
aR            GOSUB Pointer
xq8         END IF
zV6       NEXT y
yT4     NEXT x
f02   ELSE
MS4     IF POINT (207,83)<>2+Button(2,2) THEN PAINT (207,83),
        2+Button(2,2),13
ki      IF POINT (312,83)<>2+Button(1,2) THEN PAINT (312,83),
        2+Button(1,2),13
uv      IF POINT (207,136)<>2+Button(4,2) THEN PAINT (207,136
        ),2+Button(4,2),13
CF      IF POINT (312,136)<>2+Button(3,2) THEN PAINT (312,136
        ),2+Button(3,2),13
AT      FOR x=3 TO 1 STEP -1
NR6       x2=3-x
wE        FOR y=1 TO 3
Iw8         IF Display(x,y,2)<>Display(4-x,y,1) THEN
D2A           Display=Display(4-x,y,1)
gQ            Col=13
pg            GOSUB Pointer
Jc            Display=Display(x,y,2)
nZ            Col=15
sJ            GOSUB Pointer
F88         END IF
Hn6       NEXT y
G14     NEXT x
IB2   END IF
J8    ON Seite+1 GOSUB Blue,Green
kM0 RETURN
Pq  REM Routine zum Ausgeben der ersten Zeigerpositionen:
Bp  PositionPaint:
xJ2   Col=15
hX    IF Seite=0 THEN
8U4     FOR x=1 TO 3
y06       x2=x-1
GY        FOR y=1 TO 3
Xp8         Display=Display(x,y,1)
7y          GOSUB Pointer
V16       NEXT y
Uz4     NEXT x
Bu2   ELSE
cv4     FOR x=3 TO 1 STEP -1
pt6       x2=3-x
0g        FOR y=1 TO 3
h08         Display=Display(x,y,2)
F6          GOSUB Pointer
d96       NEXT y
c74     NEXT x
eX2   END IF
5h0 RETURN
A0  REM Routine zum Ausgeben der geaenderten Zeigerpositionen:
5u  PositionOutput:
1r2   IF Seite=0 THEN
So4     FOR x=1 TO 3
IK6       x2=x-1
as        FOR y=1 TO 3
XS8         IF Display(x,y,1) <> OldDisplay(x,y,1) THEN
QQA           Display=OldDisplay(x,y,1)
K4            Col=13
TK            GOSUB Pointer
JC            IF Sound1=1 THEN GOSUB Sound1
wE            Display=Display(x,y,1)
SE            Col=15
X0            GOSUB Pointer
un8         END IF
wS6       NEXT y
vQ4     NEXT x
cL2   ELSE
3M4     FOR x=3 TO 1 STEP -1
GK6       x2=3-x
p7        FOR y=1 TO 3
ro8         IF Display(x,y,2) <> OldDisplay(x,y,2) THEN
k1A           Display=OldDisplay(x,y,2)
ZJ            Col=13
iZ            GOSUB Pointer
yR            IF Sound1=1 THEN GOSUB Sound1
DW            Display=Display(x,y,2)
hT            Col=15
md            GOSUB Pointer
928         END IF
Bh6       NEXT y
Af4     NEXT x
C52   END IF
dF0 RETURN
Vf  REM Anschalten des Colour Cyclings:
1M  CyclingOn1:
uk2   MENU 3,4,1
xi    MENU 3,3,2
gH    Cycle=1
J8    ON Seite+1 GOSUB Blue,Green
kM0 RETURN
EC  REM Abschalten des Colour Cyclings:
ae  CyclingOff1:
xm2   MENU 3,3,1
8u    MENU 3,4,2
kK    Cycle=0
qF    ON Seite+1 GOSUB Blue,Green
rT0 RETURN
1U  REM Darstellung der Grafik von 18BIT mit Hilfe einiger Unte
    rroutinen:
ey  Graphic:
ek2   MOUSE STOP
7E    MENU STOP
uu    GOSUB ColourDefining
4d    GOSUB Blue
VJ    PAINT (0,0),12
fx    GOSUB Circles
55    GOSUB Frame
2X    PAINT (0,0),0
zq    GOSUB Displays
n0    GOSUB Buttons
Nv    GOSUB FlipGadget
9V    GOSUB ArrowGadgets
p6    MOUSE ON
zP    MENU ON
8k0 RETURN
AT  REM Routine zum Darstellen der Kreise fuer den Colour Cycli
    ng-Effekt:
RY  Circles:
932   FOR r=217 TO 168 STEP - 7
xK4     CIRCLE (260,109),r,r/7-20
Bf      PAINT (260,109),r/7-20
b02   NEXT r
KX    FOR r=161 TO 105 STEP - 7
2K4     CIRCLE (260,109),r,r/7-11
Gk      PAINT (260,109),r/7-11
f42   NEXT r
```





```
431 vh    FOR r=98 TO 42 STEP - 7
432 Xw4     CIRCLE (260,109),r,r/7-2
433 X1      PAINT (260,109),r/7-2
434 j82   NEXT r
435 NzO RETURN
436 y6  REM Routine zum Darstellen des Rahmens von 18BIT:
437 AM  Frame:
438 8P2   LINE (143,20)-(367,20),0
439 FC    RESTORE FrameData
440 xN    FOR x=1 TO 7
441 E94     READ a1,a2
442 5K      LINE -(a1,a2),0
443 4Z2   NEXT x
444 W80 RETURN
445 W9  REM Koordinaten fuer den Rahmen:
446 x5  FrameData:
447 7g2   DATA 460,60,460,158,367,198,143,198,60,158,60,60,143,20
448 5t0 REM Routine zum Darstellen des Gadgets zum Seitenwechseln:
449 am  FlipGadget:
450 ex2   LINE (467,89)-(508,127),14,b
451 JK    LINE (509,127)-(509,89),14
452 n4    LINE (466,90)-(507,128),13,bf
453 w0    COLOR 0,13
454 Rf    LOCATE 13,60
455 PY    PRINT "FLIP"
456 iKO RETURN
457 6x  REM Routine zum Darstellen der Aktivierungsknoepfe:
458 eC  Buttons:
459 D52   RESTORE ButtonData
460 OA    FOR x = 1 TO 4
461 YT4     READ a1,a2
462 7H      CIRCLE (a1,a2),17,13,,,.36
463 o8      PAINT (a1,a2),2,13
464 Ze2   NEXT
465 rTO RETURN
466 BD  REM Koordinaten fuer die Buttonellipsen:
467 ww  ButtonData:
468 vm2   DATA 207,83,312,83,207,136,312,136
469 kJO REM Routine zum Darstellen der Displays:
470 I1  Displays:
471 Og2   COLOR 0,1
472 Hd    FOR x=1 TO 3
473 Og4     FOR y=1 TO 3
474 hD6       xKoord=154+(x-1)*106
475 Ua        yKoord=31+(y-1)*53
476 Km        LINE (xKoord,yKoord)-STEP(42,13)
477 PD        LINE -STEP(0,24)
478 m1        LINE -STEP(-42,13)
479 w8        LINE -STEP(-42,-13)
480 ep        LINE -STEP(0,-24)
481 zW        LINE -(xKoord,yKoord)
482 rX        LINE (xKoord+10,yKoord+3)-STEP(-20,0)
483 CW        PAINT (xKoord,yKoord+2),15,0
484 EC        PAINT (xKoord,yKoord+30),13,0
485 g7        FOR a = 1 TO 12
486 tr8         LINE (xKoord,yKoord+25)-(xpos(a)+(x-1)*106,ypos(a)+
                (y-1)*53)
487 2A6       NEXT a
488 pL4     NEXT y
489 oJ2   NEXT x
490 GsO RETURN
491 PV  REM Koordinaten fuer die Abschnittslinien der Displays:
492 9x  SectorData:
493 WP2   DATA 164,34,184,40,196,50,196,62,184,72,164,78
494 3o    DATA 144,78,124,72,112,62,112,50,124,40,144,34
495 FXO REM Koordinaten fuer die Ecken der Displays:
496 DJ  SectorData2:
497 J12   DATA 154,31,196,44,196,68,154,81,112,68,112,44
498 7bO REM Routine zum Zeichnen der 8 Gadgets zum Veraendern der Z
        eigerposition:
499 1B  ArrowGadgets:
500 d62   RESTORE ArrowData
501 au    COLOR 14,0
502 aL    FOR b=1 TO 2
503 w04     REM Arrow #1:
504 gU6       READ kx,ky
505 1n        LINE (kx,ky)-STEP(58,7)
506 OG        LINE -STEP(0,-15)
507 kR        LINE -(kx,ky)
508 x9        PAINT (kx+5,ky),13,14
509 5A4     REM Arrow #2:
510 ma6       READ kx,ky
511 Mt        LINE (kx,ky)-STEP(-58,7)
```



```
512 6M        LINE -STEP(0,-15)
513 qX        LINE -(kx,ky)
514 FJ        PAINT (kx-5,ky),13,14
515 EK4     REM Arrow #3:
516 sg6       READ kx,ky
517 k5        LINE (kx,ky)-STEP(17,25)
518 VC        LINE -STEP(-35,0)
519 wd        LINE -(kx,ky)
520 Hm        LINE (kx,ky+1)-STEP(16,24)
521 kd        LINE (kx,ky+1)-STEP(-17,24)
522 8F        PAINT (kx,ky+5),13,14
523 PW4     REM Arrow #4:
524 Oo6       READ kx,ky
525 H6        LINE (kx,ky)-STEP(17,-25)
526 dK        LINE -STEP(-35,0)
527 41        LINE -(kx,ky)
528 hb        LINE (kx,ky-1)-STEP(16,-24)
529 X4        LINE (kx,ky-1)-STEP(-17,-24)
530 OX        PAINT (kx,ky-5),13,14
531 mv2   NEXT b
532 wYO RETURN
533 h1  REM Koordinaten fuer die Arrow Gadgets:
534 DP  ArrowData:
535 kU2   DATA 367,10,143,10,483,158,483,60
536 Hf    DATA 367,208,143,208,37,158,37,60
537 ksO REM Erzeugung von wirklich zufaelligen Werten fuer die Disp
        lays:
538 Or  New1:
539 s12   RANDOMIZE TIMER
540 Nj    FOR x=1 TO 3
541 Um4     FOR y=1 TO 3
542 Y16       FOR z=1 TO 2
543 hJ8         OldDisplay(x,y,z)=Display(x,y,z)
544 a6          Display(x,y,z)=INT(1+12*RND)
545 mJ6       NEXT z
546 1H4     NEXT y
547 kF2   NEXT x
548 c6    FOR x=1 TO 3 STEP 2
549 j94     FOR y=1 TO 3 STEP 2
550 C16       Display(x,y,2)=12-Display(x,y,1)
551 Rd        IF Display(x,y,2)=0 THEN Display(x,y,2)=12
552 rN4     NEXT y
553 qL2   NEXT x
554 dB    GOSUB PositionOutput
555 JvO RETURN
556 Qn  REM Routine zum Abspeichern des momentanen Spielzustandes u
        nter einem eingegebenen Namen:
557 GX  SaveAs1:
558 ef2   WINDOW 2,"Spielstand Abspeichern/RETURN für Cancel",(0,0)
          -(511,8),2,1
559 o6    COLOR 0,1
560 a1    LOCATE 1,1
561 C9    LINE INPUT "Filename: ";Filename$
562 Gw    WINDOW CLOSE 2
563 Ov    IF Filename$<>"" THEN Oldname$=Filename$ : GOSUB Save1
564 1E    IF Filename$="" THEN Filename$=Oldname$
565 T50 RETURN
566 tP  REM Routine zum Abspeichern des momentanen Spielstandes unt
        er dem alten Dateinamen:
567 XE  Save1:
568 9f2   IF Filename$="" THEN
569 k54     BEEP : GOSUB SaveAs1
570 nW2   ELSE
571 x14     OPEN Filename$ FOR OUTPUT AS 1
572 tF6     FOR x=1 TO 3
573 OI8       FOR y=1 TO 3
574 4HA         FOR z=1 TO 2
575 70C           PRINT #1,Display(x,y,z)
576 HoA         NEXT z
577 Gm8       NEXT y
578 Fk6     NEXT x
579 3Q      FOR x=1 TO 4
580 4L8       FOR y=1 TO 2
581 1aA         PRINT #1,Button(x,y)
582 Lr8       NEXT y
583 Kp6     NEXT x
584 fH      PRINT #1,Seite
585 Fz4     CLOSE 1
```

Listing. »Gehirntraining« mit »18BIT« ist eine spannende Angelegenheit. Bitte mit dem Checksummer (Seite 143) eingeben (Fortsetzung).

```
  END IF
RETURN
REM Routine zum Abspeichern des momentanen Spielstandes unt
er einem voreingestellten Namen:
Quicksave1:
  Filename$="18BIT.quick"
  GOSUB Save1
  Filename$=Oldname$
RETURN
REM Routine zum Laden eines abgespeicherten Spielstandes un
ter einem eingegebenen Namen:
Load1:
  WINDOW 2,"Spielstand Laden/RETURN für Cancel",(0,0)-(511,
  8),2,1
  COLOR 0,1
  LOCATE 1,1
  LINE INPUT "Filename: ";Filename$
  WINDOW CLOSE 2
  IF Filename$<>"" THEN
    Oldname$=Filename$
    File=1
    GOSUB Quickload1
    File=0
  ELSE
    Filename$=Oldname$
  END IF
RETURN
REM Laden des mit Quicksave gespeicherten Spielstandes:
Quickload1:
  Col=13
  IF Seite=0 THEN
    FOR x=1 TO 3
      x2=x-1
      FOR y=1 TO 3
        Display=Display(x,y,1)
        GOSUB Pointer
      NEXT y
    NEXT x
  ELSE
    FOR x=3 TO 1 STEP -1
      x2=3-x
      FOR y=1 TO 3
        Display=Display(x,y,2)
        GOSUB Pointer
      NEXT y
    NEXT x
  END IF
  IF File=1 THEN Filename1$=Filename$ ELSE Filename1$="18BI
  T.quick"
  OPEN Filename1$ FOR INPUT AS 1
    FOR x=1 TO 3
      FOR y=1 TO 3
        FOR z=1 TO 2
          INPUT #1,Display(x,y,z)
        NEXT z
      NEXT y
    NEXT x
    FOR x=1 TO 4
      FOR y=1 TO 2
        INPUT #1,Button(x,y)
      NEXT y
    NEXT x
    INPUT #1,Seite
  CLOSE 1
  IF Seite=0 THEN
    IF POINT (207,83)<>2+Button(1,1) THEN PAINT (207,83),
    2+Button(1,1),13
    IF POINT (312,83)<>2+Button(2,1) THEN PAINT (312,83),
    2+Button(2,1),13
    IF POINT (207,136)<>2+Button(3,1) THEN PAINT (207,136
    ),2+Button(3,1),13
    IF POINT (312,136)<>2+Button(3,1) THEN PAINT (312,136
    ),2+Button(4,1),13
  ELSE
    IF POINT (207,83)<>2+Button(2,2) THEN PAINT (207,83),
    2+Button(2,2),13
    IF POINT (312,83)<>2+Button(1,2) THEN PAINT (312,83),
    2+Button(1,2),13
    IF POINT (207,136)<>2+Button(4,2) THEN PAINT (207,136
    ),2+Button(4,2),13
    IF POINT (312,136)<>2+Button(3,2) THEN PAINT (312,136
    ),2+Button(3,2),13
  END IF
  GOSUB PositionPaint
  ON Seite+1 GOSUB Blue,Green
RETURN
REM Ein versuchter Abbruch durch Break wird unterbunden:
BreakTrapping:
  BEEP
RETURN
REM Routine zum Erzeugen eines Sounds beim Verstellen der D
isplays:
Sound1:
  FOR a1=4 TO 1 STEP -1
    SOUND 100,.1,a1*a1*16-1
    FOR x1=1 TO 20
      SOUND WAIT
    NEXT x1
    SOUND RESUME
  NEXT a1
RETURN
REM Anschalten des Sounds:
SoundOn1:
  MENU 3,2,1
  MENU 3,1,2
  Sound1=1
RETURN
REM Abschalten des Sounds:
SoundOff1:
  MENU 3,1,1
  MENU 3,2,2
  Sound1=0
RETURN
REM Start einer Sequenz:
SeqStart1:
  Seq=1
  WINDOW 2,"Sequenz Abspeichern/RETURN fÜr Cancel",(0,0)-(5
  11,8),2,1
  COLOR 0,1
  LOCATE 1,1
  LINE INPUT "Sequence-/Filename: ";Seqname$
  WINDOW CLOSE 2
  IF Seqname$="" THEN
    Seq=0
  ELSE
    OPEN Seqname$ FOR OUTPUT AS 2
  END IF
RETURN
REM Stop einer Sequenz:
SeqStop1:
  Seq=0
  CLOSE 2
RETURN
REM Laden und Zeigen einer Sequenz:
SeqShow1:
  WINDOW 2,"Sequenz Laden/RETURN für Cancel",(0,0)-(511,8),
  2,1
  COLOR 0,1
  LOCATE 1,1
  LINE INPUT "Sequence-/Filename: ";Seqname$
  WINDOW CLOSE 2
  IF Seqname$<>"" THEN
    IF Cycle=1 THEN Cycle1=1 : Cycle=0
    Seite2=Seite
    OPEN Seqname$ FOR INPUT AS 2
      WHILE NOT EOF(2)
        FOR x=1 TO 4
          FOR y=1 TO 2
            INPUT #2,Button(x,y)
          NEXT y
        NEXT x
        INPUT #2,Seite1
        INPUT #2,Hebel
        INPUT #2,Richtung
        IF Seite1=Seite THEN Seite=1-Seite
        GOSUB ScreenChange
        GOSUB DisplayPosition
      WEND
    CLOSE 2
    IF Cycle1=1 THEN Cycle=1
    Cycle1=0
    IF Seite<>Seite2 THEN GOSUB ScreenChange
  END IF
RETURN
REM Routine zum Abspeichern der einzelnen Sequenzenabschnit
te:
```



```
Sequence:
  FOR x=1 TO 4
    FOR y=1 TO 2
      PRINT #2,Button(x,y)
    NEXT y
  NEXT x
  PRINT #2,Seite
  PRINT #2,Hebel
  PRINT #2,Richtung
RETURN
REM Unterprogramm zum Zeichnen der Zeiger auf dem Display:
Pointer:
  COLOR Col,0
  xKoord=154+x2*106
  yKoord=56+(y-1)*53
  IF Display=12 THEN
    xk1=xpos(12)+x2*106 : yk1=ypos(12)+(y-1)*53
    xk2=xpos(1)+x2*106 : yk2=ypos(1)+(y-1)*53
  ELSE
    xk1=xpos(Display)+x2*106 : yk1=ypos(Display)+(y-1)*53
    xk2=xpos(Display+1)+x2*106 : yk2=ypos(Display+1)+(y-1)*
    53
  END IF
  AREA (xKoord,yKoord)
  AREA (xk1,yk1)
  IF Display/2=INT(Display/2) AND Display<>12 THEN
    AREA(xpos2(Display/2+1)+x2*106,ypos2(Display/2+1)+(y-1)
    *53)
  END IF
  AREA (xk2,yk2)
  AREA (xKoord,yKoord)
  AREAFILL
  COLOR 0,1
  LINE (xKoord,yKoord)-(xk1,yk1)
  IF Display/2=INT(Display/2) AND Display<>12 THEN
    LINE -(xpos2(Display/2+1)+x2*106,ypos2(Display/2+1)+(y-
    1)*53)
  END IF
  LINE -(xk2,yk2)
  LINE -(xKoord,yKoord)
RETURN
ENDE:
(C) 1989 M&T
```

Listing. Gehirntraining mit »18 BIT« ist eine spannende Angelegenheit. Bitte mit dem Checksummer (Seite 143) eingeben (Schluß).

SPIEL, SPASS & SPANNUNG

Spielen macht Spaß. Vor allem, wenn ein Programm von der Bedienung, Geschwindigkeit und Grafik her absolute Spitze ist: Wie unser »Buh-Au-Construction-Set«.

Eimerkette« oder »Chain Reaction« (Ausgabe 11/87, Seite 66), unter diesen Namen wurden Listings veröffentlicht, die die Grundlage für unser Programm des Monats, das »Buh-Au-Construction-Set« (kurz »BACS«) bilden. Das heißt jedoch nicht, daß dieses Spiel nur eine reine Umsetzung der anderen Listings ist, vielmehr wurde nur das Spielprinzip übernommen. Falls Sie »Eimerkette« nicht kennen, eine kurze Spielbeschreibung: Ziel des Spiels ist es, den Gegner von einem schachbrettartig eingeteilten Spielfeld zu verdrängen. Zu diesem Zweck kann jeder der Spieler Felder belegen. Es darf jedoch pro Spielzug nur jeweils ein Feld belegt werden. Gesetzt werden darf nur auf noch nicht belegte oder schon mit der eigenen Farbe besetzte Felder. Im letzteren Fall erhöht sich der Feldinhalt um eins. Da ein Feld aber nicht unendlich viel aufnehmen kann, »schwappt« der Inhalt über, sobald der Feldindex gleich der Anzahl der horizontalen und vertikalen Nachbarfelder ist (also maximal vier). Dabei wird jedem der benachbarten Felder die Farbe des »übergelaufenen« Feldes zugeordnet und um eins erhöht, was natürlich einen erneuten »Überlauf« zur Folge haben kann. So kann man durch geschicktes Erhöhen der Felder ganze Ketten aufbauen. Soweit zum Spielprinzip von »Eimerkette«. Wie bereits erwähnt, ist unser Spiel eine Weiterentwicklung. Bei dem Spiel »Buh-Au-Construction-Set« sind noch einige Elemente hinzugekommen:

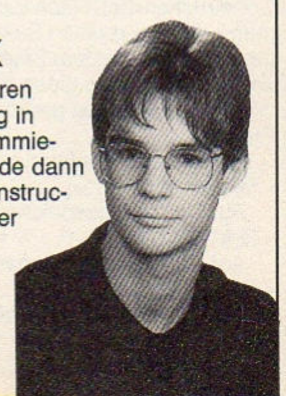
- Das Spielfeld ist maximal 15 x 15 Felder groß (im Gegensatz zu 7 x 7 bei Eimerkette).
- Es können eigene Spielfelder entworfen werden, so daß man nicht nur auf den mit der Zeit langweiligen, rechteckigen Spielfeldern spielen muß. Diese Felder können auf Diskette abgespeichert werden.
- Bis zu drei Spieler können gleichzeitig gegeneinander antreten, was den Spielreiz deutlich erhöht. Bei nur einem Spieler übernimmt der Amiga die Rolle des Gegners.
- Die Bedenkzeit für einen Zug wurde begrenzt. Die maximale Zeit kann von 5 bis 30 Sekunden in 5-Sekunden-Schritten variiert werden.
- Nach Spielende werden für jeden Spieler Punkte berechnet und in einer High-Score-Liste auf Disk gespeichert. Für die Verlierer wird eine »Low-Score-Liste« geführt, die ebenfalls gespeichert wird.
- Das ganze Programm ist mausgesteuert. Die Tastatur wird nur zur Eingabe der Spielernamen, beim Speichern und zum Aktivieren einiger Sonderfunktionen wie »Titelaufbau aus/ein«, »Pause« und »Spiel abbrechen« benötigt.

Diese Erweiterungen bewegten die Autoren dazu, den Namen des Spiels zu ändern. Der Name »Buh-Au-Construction-Set« hat keinen tieferen Sinn und soll nicht zum Nachdenken verleiten. Den



MARKUS BONER, KLAUS SCHRECK

Die »Computerlaufbahn« der zwei Programmautoren verlief ähnlich: Ende 1984 fanden sie den Einstieg in die Computerwelt über den C 64. Nach der Programmierung in Basic folgte Assembler. Vor ca. 1,5 Jahren wurde dann der Traumcomputer Amiga angeschafft. »Buh-Au-Construction-Set« ist das erste größere Projekt. Wegen der noch fehlenden Kenntnis des 68000-Assemblers wurde es vernünftigerweise in Basic programmiert. Das »Buh-Au-Construction-Set« wurde Programm des Monats in der AMIGA-Ausgabe 5/89. Wir wünschen den beiden noch weitere erfolgreiche Projekte.

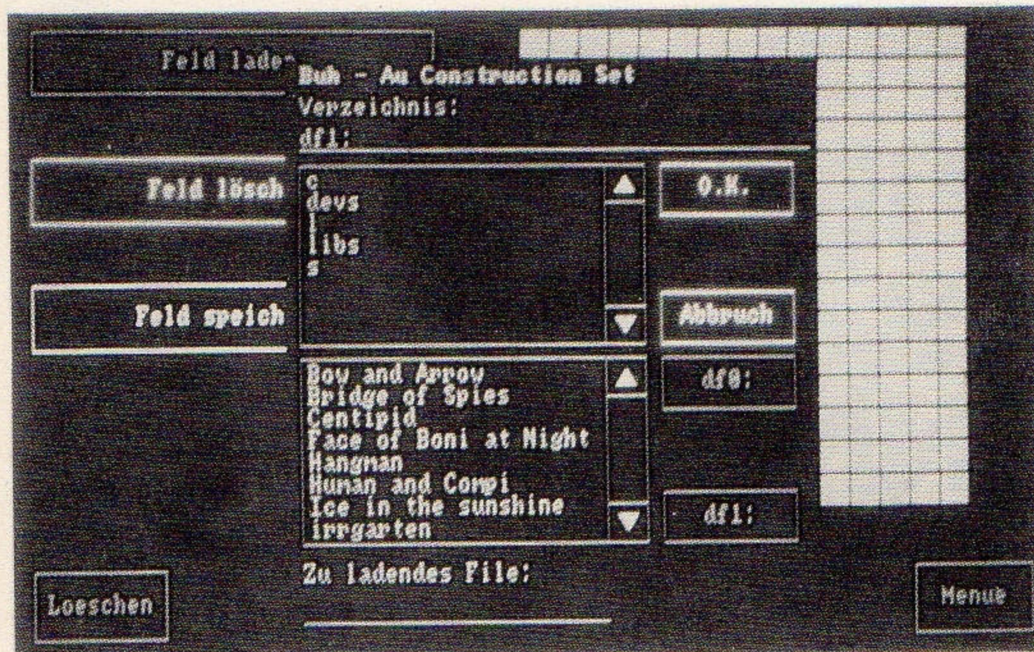


Autoren gefiel aber die Abkürzung so gut. Auf der Diskette müssen sich folgende Dateien befinden:
- Buh-Au_CS
- intuition.bmap
- exec.bmap
- graphics.bmap
- dos.bmap

Falls Sie die ».bmap«-Dateien nicht auf Ihrer Extras-Diskette im Verzeichnis »BasicDemos« finden, müssen Sie sie selbst generieren. Eine genaue Anleitung, wie dies funktioniert, finden Sie in der Ausgabe 1/89 des AMIGA-Magazins auf Seite 147.

Da das Programm sehr »speicherfressend« ist, sollten Sie darauf achten, daß beim Aufruf noch mindestens 380 KByte Speicher frei sind. Um das Programm zu starten, verwenden Sie das Ladeprogramm (siehe Listing 1). Beendet werden sollte das Programm nur über die eingebaute »QUIT« Funktion.

Nach dem Starten des Programms werden zunächst die .bmap-Dateien und das Inhaltsverzeichnis der Diskette im internen Laufwerk geladen. Anschließend sucht das Programm nach den High-Scores und generiert eine eigene High-Score-Liste, falls keine auf Disk gefunden werden. Diese Diskettenoperationen dauern natürlich etwas (man hat genug Zeit, zur Kaffeemaschine zu kommen). Danach erscheint das Titelbild, das gleichzeitig als Hauptmenü fungiert und im folgenden auch so bezeichnet wird. Da auch der Titelaufbau recht lange dauert, kann er durch Druck auf <F1> an- und ausgeschaltet werden (die Einstellung wird erst beim nächsten Aufbau des Hauptmenüs berücksichtigt). Nun zu den einzelnen Menüpunkten, die durch einfaches Anklicken der Schalter angewählt werden.



Spiel, Spaß und Spannung für viele Stunden bietet das »Buh-Au-Construction-Set«

■ Spielfeld konstruieren

Bevor ein Spiel gestartet werden kann, muß erst ein Spielfeld konstruiert werden. Zu diesem Zweck begibt man sich in das Construction-Set. Nach kurzer Wartezeit ist man im Konstruktionsmenü. Auf der rechten Seite des Bildschirms ist eine 15 x 15 Felder große Matrix, die dem Spielfeld entspricht. Durch Anklicken einzelner Felder können diese entweder gesetzt oder gelöscht werden. Wird die Maustaste gedrückt gehalten, setzt BACS kontinuierlich Punkte.

Die anderen Funktionen haben folgende Bedeutungen:
- Feld laden:
  Auf Diskette gespeicherte Felder werden geladen.
- Feld löschen:
  Auf Diskette gespeicherte Felder löschen.
- Feld speichern:
  Feld auf Disk abspeichern
- Loeschen:
  Spielfeld löschen
- Fuellen:
  Spielfeld ganz ausfüllen
- Tester:
  Dieser Menüpunkt muß aktiviert werden, bevor gespielt werden kann. Das Spielfeld wird nach Aufruf dieses Punktes auf seine Spielbarkeit hin überprüft. Spielbare Felder dürfen keine Einerfelder enthalten, das heißt, jedes Feld muß mindestens zwei Nachbarfelder haben. Weiterhin muß das Spielfeld in einem durchgehen, das heißt, es darf nur ein abgeschlossenes Spielfeld vorhanden sein. Der Tester meldet solche unkorrekten Spielfelder entweder mit »Einerfeld vorhanden« oder mit »Kein korrektes Spielfeld«. Zugelassene Felder werden mit »Spielfeld ist zugelassen« quittiert.
- Menue:
  Rücksprung ins Hauptmenü.

Für sämtliche Diskettenoperationen wird ein Filerequester, der selbständig ein angeschlossenes Laufwerk DF1: erkennt, verwendet. Dessen Bedienung wird kurz erläutert: Oben wird in Fettdruck der Diskettenname angegeben, darunter das aktuelle Verzeichnis. Im oberen der beiden Kästen sind die Unterverzeichnisse aufgelistet, in die durch einfaches Anklicken verzweigt werden kann (mit gedrückter Maustaste kann aufwärts oder abwärts »gefahren« werden). Durch Anklicken der kleinen Pfeile kann der Anzeigebereich verschoben werden. Das gleiche gilt für den »File-Kasten« darunter. Jeder Lade-, Speicher- oder Löschvorgang muß durch Anklicken von »O.K.« bestätigt werden. Zur Eingabe des Savefile-Namens muß unter »Zu speicherndes File« geklickt werden, worauf der Cursor erscheint. Nachdem ein Spielfeld entworfen oder geladen wurde, müssen noch die Spieleinstellungen vorgenommen werden. Nach Anklicken der entsprechenden Box im Hauptmenü können folgende Einstellungen vorgenommen werden:
- Die maximale Denkzeit für einen Zug (einfach in der oberen Reihe die gewünschte Zeit anklicken).
- Die Anzahl der Spieler. Die Spielernamen werden nach der Auswahl der Spieleranzahl automatisch abgefragt. Durch Anklicken von »Menü« gelangen Sie wieder in das Hauptmenü. Jetzt kann das Spiel gestartet werden. Sollte irgendeine Operation nicht korrekt ausgeführt worden sein, so meldet der Computer, was noch gemacht werden muß.

Während des Spiels ist der Bildschirm wie folgt aufgeteilt: Die rechte Hälfte beinhaltet das Spielfeld. Auf der linken Seite wird angezeigt, wer gerade am Zug ist. Unter dieser Anzeige sind die Spielernamen aufgeführt. Hinter diesen wird die vom Spieler jeweils schon in Anspruch genommene Denkzeit angegeben. In der Mitte oben wird angezeigt, wieviel Zeit dem Spieler noch für seinen Zug verbleibt. Die letzten drei Sekunden werden durch Pieptöne begleitet. Durch Druck auf die linke Maustaste wird das Spiel und die Zeit des ersten Spielers gestartet. Um auf ein Feld zu setzen, wird einfach mit der Maus das entsprechende Feld angeklickt. Falls das Feld nicht besetzt werden darf (weil es schon ein anderer besetzt hat), wird dies durch einen Pieps mitgeteilt. Durch Druck auf <F10> läßt sich das Spiel in den Pausemodus bringen. Zum Weiterspielen einfach die linke Maustaste drücken. Wenn man keine Lust hat, das Spiel zu Ende zu spielen, kann es durch <CTRL Q> abgebrochen werden. Zur Sicherheit wird das Spielfeld ausgeblendet (auch bei Pause) und eine Sicherheitsabfrage durchgeführt. Nach Spielende wird der Gewinner bekanntgegeben und die High- und Low-Score-Liste gezeigt. Falls ein Spieler in eine der beiden Listen aufgenommen wurde, wird diese automatisch auf Diskette gespeichert. Ins Hauptmenü gelangt man durch einen Druck auf die linke Maustaste. Während der Benutzung des Programms kommt es vor, daß der Computer eine Meldung zu machen hat. Sie erscheint rot und zentriert auf dem Bildschirm. Um das Programm fortzusetzen, drücken Sie einfach die linke Maustaste.

Der letzte Hauptmenüpunkt ist »QUIT«. Nach Anklicken der Box wird noch einmal eine Sicherheitsabfrage durchgeführt.

Wenn Sie außer Spielen auch noch das Programmieren im Kopf haben, sollten Sie sich das Programm etwas genauer ansehen. Durch die Benutzung des Betriebssystems ist das Programm komfortabel und schnell. Die Auswahl einer Datei sollte immer über einen Filerequester geschehen, wie BACS es vormacht. Fehleingaben sind dadurch vermeidbar. Außerdem muß der Benutzer nicht raten, wie die Datei heißt. Überprüfungen, wie BACS sie vornimmt, beispielsweise kein Spielstart ohne vorherige Voreinstellungen, sind beispielhaft. Übrigens finden Sie auf unserer Programmservice-Diskette 15 fertige Spielfelder. Das heißt viele Stunden Spiel, Spaß und Spannung…

Markus Boner/Klaus Schreck/kn

Programmname:	Lade_Buh-Au_CS
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2
Bemerkung:	Startprogramm

```
Programmautor: Markus Boner, Klaus Schreck
-----------------------------------------

W80 CLEAR,45000&
Xn  LOAD"buh-au_cs",r
(C) 1989 M&T
```

Listing 1. Das Ladeprogramm zum »Buh-Au-Construction-Set« lädt das Hauptprogramm

Programmname:	Buh-Au_CS
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2

```
Programmautor: Markus Boner, Klaus Schreck
-----------------------------------------

yFO LIBRARY"graphics.library"
aM  LIBRARY"exec.library"
UY  LIBRARY"dos.library"
CZ  LIBRARY"intuition.library"
UL  DECLARE FUNCTION AllocMem& LIBRARY
Oc  DECLARE FUNCTION Examine& LIBRARY
3q  DECLARE FUNCTION ExNext& LIBRARY
f9  DECLARE FUNCTION Lock& LIBRARY
qX  DECLARE FUNCTION UnLock& LIBRARY
8U  DECLARE FUNCTION OpenDevice& LIBRARY
a6  DECLARE FUNCTION CloseDevice& LIBRARY
QD  ex&=2^17:info&=AllocMem&(252&,ex&)
hp  a$="trackdisk.device"+CHR$(0)
Qx  ex&=OpenDevice&(SADD(a$),1&,info&,0):df1%=ex&<>0
ph  IF NOT df1% THEN
Uv2   ex&=CloseDevice&(info&)
D60 END IF
Tc  RANDOMIZE TIMER
Z5  DEF FN Maus=PEEK(12574721&) AND 64
tl  DEFINT a,e,i,x,y,g,k,h,j,z
he  DIM sbc%(3),min%(3),sec%(3),feld%(14,14),init%(14,14),feld
    farbe%(14,14),dir$(100)
kI  DIM stapel%(225,2),wert%(3),spielername$(3),punkte%(3),nam
    en$(3),file$(100)
Vy  DIM hsc%(13),hsc$(13),lsc%(13),lsc$(13),einr(100,1)
Zt  drive$=":":GOSUB LiesDiskName
kN  UrDisk$=file$(0):a$=UrDisk$+":SCORES.bcs"+CHR$(0)
Qe  ex&=Lock&(SADD(a$),-2):ScoreFehler%=ex&=0
I7  ex&=UnLock&(ex&):disk$="df0:":drive$=disk$
od  GOSUB LeseDir:GOSUB HighScores
SH  WINDOW CLOSE 1:SCREEN 1,640,255,3,2:WINDOW 2,,,16,1
Q9  Farbenloeschen
B1  ON MOUSE GOSUB MouseTest
FQ  ON TIMER(1) GOSUB ZeitAuswertung
dN  GOSUB Loeschen:title%=-1:flag%=1:GOSUB Aufbau1:MOUSE ON
cS  Endlos:
1h  IF komp AND zahl=2 THEN GOSUB Computer
gE  a$=INKEY$
kd  IF a$<>"" THEN
DR2   IF a$=CHR$(129) AND flag%=1 THEN
L34     MOUSE OFF:SOUND 800,2,255,0:title%=NOT title%
Yx      Wertung$="Titelaufbau "
FH      IF title% THEN Wertung$=Wertung$+"ein" ELSE Wertung$=W
        ertung$+"aus"
vl      Meldung Wertung$,2:MOUSE ON
dW2   END IF
Ec    IF a$=CHR$(17) AND flag%=24 THEN
3i4     TIMER OFF:MOUSE OFF
j5      WINDOW 4,,(300,30)-(570,210),0,1
cm      GOSUB Sicher:WINDOW CLOSE 4
j7      IF ja THEN
936       CLS:flag%=1
Mz        GOSUB NeustartInitialisierungsRoutine
eo        IF komp THEN komp=0:spanz%=1
ez        GOSUB Aufbau1
SB5      ELSE
616       flag%=24:TIMER ON
pi4     END IF
zG      MOUSE ON
rk2   END IF
S3    IF a$=CHR$(138) AND flag%=24 THEN
m74     TIMER OFF:MOUSE OFF:WINDOW 4,,(300,30)-(570,210),0,1
gq      COLOR 5,0:LOCATE 7,8:Printe "- P - A - U - S - E -"
hl      COLOR 4,0:LOCATE 10,7:Printe "Zum Weiterspielen bitte"
PD      LOCATE 12,4:Printe "die linke Maustaste dr"+CHR$(252)+
        "cken!!"
OJ      CALL WaitForClick :WINDOW CLOSE 4:MOUSE ON:TIMER ON
yr2   END IF
S1    WHILE INKEY$<>"":WEND
OtO END IF
Ea  IF MOUSE(0)=0 THEN Endlos ELSE Endlos
t7  MouseTest:
oe  mx=MOUSE(1):my=MOUSE(2)
yc  IF flag%=1 THEN RESTORE Daten1
Ta  IF flag%>2 AND flag%<7 OR flag%=20 THEN RESTORE Daten2
pm  IF flag%=11 OR flag%=22 THEN RESTORE Daten4
23  IF flag%=13 OR flag%=23 THEN RESTORE Daten5
FK  IF flag%=15 OR flag%=24 THEN RESTORE Daten6
Vi  IF flag%=-3 OR flag%=-2 THEN RESTORE Daten7
UX  j=flag%
eI  Neu:
MZ  READ x1
4Q  IF x1=-1 THEN flag%=j:RETURN
dp  READ y1,x2,y2,flag%,farbe%,wort$
Ms  IF mx>=x1 AND mx<=x2 AND my>=y1 AND my<=y2 THEN
xi2   MOUSE OFF
Wg    IF flag%<20 THEN
vJ4     Zeichnen2 x1,y1,x2,y2,farbe%,wort$
1Y      IF flag%<-1 THEN GOSUB ja:RETURN
cR      ON flag% GOSUB Aufbau1,Quit,Aufbau2,Loeschen,Checker,F
        uellen,Out,Out,Out,,Aufbau3,Anklicken2,Spielerauswahl,A
        bbruch,Aufbau4
Ul      MOUSE ON
m0      RETURN
213    ELSE
A14     ON (flag%-19) GOSUB Anklicken,,Anklicken2,SpielerEinga
        be,Setzen
Yp      MOUSE ON
qS      RETURN
RK2   END IF
SLO END IF
WJ  GOTO Neu
Cx  Daten1:
JG  DATA 60,140,300,172,3,2,Spielfeld konstruieren,340,140,580
    ,172,11,3,Spieleinstellung
Sn  DATA 60,184,300,216,15,4,Spiel starten,340,184,580,216,2,5
    ,Q U I T,-1
H3  Daten2:
C2  DATA 300,0,570,180,20,0,,8,200,88,224,4,6,Loeschen,360,200
    ,440,224,5,4,Tester,536,200,608,224,1,3,Menue
8X  DATA 8,0,248,24,7,7,Feld laden,8,96,248,120,8,5,Feld speic
    hern,176,200,248,224,6,2,Fuellen
Va  DATA 8,49,248,73,9,4,Feld löschen,-1
NA  Daten3:
pw  DATA 220,38,300,57,7,4,O.K.,220,86,300,105,8,4,Abbruch,220
    ,110,300,129,9,3,"df0:"
2r  DATA 220,159,300,177,10,3,"df1:",4,38,186,105,1,,,186,38,2
    10,52,2,,,186,91,210,105,3,,,4,110,186,177
xV  DATA 4,,,186,110,210,124,5,,,186,163,210,177,6,,,4,194,186
```

Listing 2. Ein Spiel mit »allem Drum und Dran«: »Buh-Au-Construction-Set«. Bitte mit dem Checksummer (Seite 143) eingeben (Fortsetzung).

```
,205,11,,,-1
Daten4:
DATA 80,0,160,24,22,2,05-Sek,160,0,240,24,22,2,10-Sek,240,0,320,24,22,2,15-Sek,320
DATA 0,400,24,22,2,20-Sek,400,0,480,24,22,2,25-Sek,480,0,560,24,22,2,30-Sek
DATA 200,48,440,72,13,5,Ein Spieler,200,104,440,128,13,5,Zwei Spieler,200,160,440,184,13,5,Drei Spieler
DATA 552,200,624,224,1,4,Menue,8,200,80,224,1,4,Menue,-1
Daten5:
DATA 80,72,160,96,14,4,Okay,4,6,236,18,23,3,,4,30,236,42,23,4,,4,54,236,66,23,5,,-1
Daten6:
DATA 300,30,570,210,24,,,-1
Daten7:
DATA 16,26,63,44,-2,5,JA,168,26,216,44,-3,3,NEIN,-1
TitelDaten:
DATA "******    **   **   **   **         *****    **   **"
DATA "**   **   **   **   **   **        **   **   **   **"
DATA "**   **   **   **   **   **        **   **   **   **"
DATA "******    **   **   ******* ***  *******    **   **"
DATA "**   **   **   **   **   **        **   **   **   **"
DATA "**   **   **   **   **   **        **   **   **   **"
DATA "******     *****    **   **        **   **    ***** "
DATA 58
DATA 40,80,5,40,93,5,42,79,2,42,105,3,75,80,5,75,93,5,77,79,2,77,105,3,101
DATA 80,1,101,93,1,115,80,5,115,93,5,119,80,6,132,93,6,148,80,1,148,93,1,163
DATA 79,2,161,80,5,163,92,4,187,93,1,163,105,3,200,79,2,225,79,2,224,80,1
DATA 224,93,1,263,80,5,265,79,2,263,93,5,289,80,1,265,92,3,275,93,6,303,80,5
DATA 303,93,5,305,105,3,329,80,1,329,93,1,343,80,5,343,93,5,345,79,2,345,105
DATA 3,382,79,2,407,79,2,406,80,1,406,93,1,445,80,1,445,93,1,459,80,5,459,93
DATA 5,461,79,2,461,105,3,485,80,1,485,93,1,499,80,5,499,93,5,503,80,6,516
DATA 93,6,532,80,1,532,93,1,2,79,2,0,80,5,2,92,4,26,93,1,2,105,3,43,80,5
DATA 43,93,5,45,79,2,45,92,4,45,105,3,83,79,2,108,79,2,107,80,1,107,93,1
Aufbau1:
CLS
RESTORE Daten1
call Farbenloeschen :call Zeichnen (4):COLOR 6,0:Gotoxy 184,240
Printe CHR$(169)+" in 1988 by M. Boner & K. Schreck"
IF title% THEN
  RESTORE TitelDaten:COLOR 6,0
  FOR i=1 TO 7
    READ a$
    FOR j=1 TO LEN(a$)
      IF MID$(a$,j,1)="*" THEN GOSUB Viereck
    NEXT
  NEXT
  COLOR 7,0:READ j
  FOR k=1 TO j:READ x,y,i:x=x+33:y=y-10:GOSUB Led:NEXT
  FOR k=1 TO 14:READ x,y,i:x=x+254:y=y+27:GOSUB Led:NEXT
END IF
call Farbensetzen
RETURN
Aufbau2:
CLS:RESTORE Daten2:call Farbenloeschen:Zeichnen 8
FOR y=0 TO 14
  FOR x=0 TO 14
    IF feld%(x,y)>0 THEN feld%(x,y)=0
    GOSUB Kaestchen
  NEXT
NEXT
COLOR 6,0:call Farbensetzen
RETURN
Aufbau3:
CLS:RESTORE Daten4:call Farbenloeschen:Zeichnen 11
IF x7 THEN Zeichnen2 x7,y7,x8,y8,2,hwort$
call Farbensetzen
RETURN
Aufbau4:
flag%=24
Neusetzen x1,y1,x2,y2,farbe%,wort$
IF anzahl=0 OR zeiger<>1 OR spanz%=0 OR zeit=0 THEN
  flag%=1
  IF anzahl=0 THEN Meldung "Kein Spielfeld vorhanden",4:RETURN
  IF zeiger<>1 THEN Meldung "Bitte erst den Tester aktivieren",4:RETURN
  IF spanz%=0 THEN Meldung "Spielereinstellung vornehmen",4:RETURN
  IF zeit=0 THEN Meldung "Zeiteinstellung vornehmen",4:RETURN
END IF
ERASE feldfarbe%:DIM feldfarbe%(14,14)
zahl=0:CLS:call Farbenloeschen:LINE (300,30)-(570,210),7,bf
FOR y=0 TO 14
  FOR x=0 TO 14
    IF feld%(x,y)<>-1 THEN
      x3=300+x*18:x4=318+x*18:y3=30+y*12:y4=42+y*12
      LINE (x3,y3)-(x4,y4),6,b
      LINE (x3+1,y3+1)-(x4-1,y4-1),2,bf
    END IF
  NEXT
NEXT
FOR i=1 TO spanz%
  sbc%(i)=0
  IF spielername$(i)="" THEN spielername$(i)="Spieler"+STR$(i):sbc%(i)=-1
NEXT
IF spanz%=1 THEN spanz%=2:komp=-1:spielername$(2)="AMIGA"
hz=zeit:COLOR 3,0
LOCATE 10,2:Printe spielername$(1)+" ist am Zug."
LOCATE 2,31:Printe "Denkzeit "+CHR$(252)+"brig: ":PRINT USING"##";hz
COLOR 6,0:LOCATE 13,2:Printe "Verbrauchte Zeiten:"
FOR i=1 TO spanz%
  min%(i)=0:sec%(i)=0:COLOR i+2,0
  LOCATE 13+2*i,2:Printe spielername$(i)
  LOCATE 13+2*i,22:Printe " : 00.00"
NEXT
call Farbensetzen
PALETTE 6,.8,1,.8:PALETTE 2,.2,.2,.7:PALETTE 7,.3,.3,.3
zahl=1
Meldung "Zum Spielbeginn Maustaste dr"+CHR$(252)+"cken",4
TIMER ON
RETURN
Viereck:
x=57+j*10:y=10+i*5:LINE (x,y)-(x+7,y+3),,bf
RETURN
Led:
IF i=1 THEN
  LINE (x,y)-(x,y+11):LINE (x-2,y+1)-(x,y+10),,bf
 ELSE
  IF i=2 THEN
    LINE (x,y)-(x+22,y):LINE (x+2,y+1)-(x+20,y+1)
   ELSE
    IF i=3 THEN
      LINE (x,y)-(x+22,y):LINE (x+2,y-1)-(x+20,y-1)
     ELSE
      IF i=4 THEN
        LINE (x,y)-(x+22,y):LINE (x+2,y-1)-(x+20,y+1),,bf
       ELSE
        IF i=5 THEN
          LINE (x,y)-(x,y+11):LINE (x,y+1)-(x+2,y+10),,bf
         ELSE
          IF i=6 THEN
            LINE (x,y)-(x+11,y+11):LINE (x+2,y)-(x+12,y+10)
          END IF
        END IF
      END IF
    END IF
  END IF
END IF
RETURN
Setzen:
IF POINT(mx,my)=7 OR POINT(mx,my)=6 THEN RETURN
x=(mx-300)\18:y=(my-30)\12
Setzen2:
IF feldfarbe%(x,y)=0 OR feldfarbe%(x,y)=zahl+2 THEN
  TIMER OFF:SOUND 800-zahl*100,2,255,0
  feldfarbe%(x,y)=zahl+2:feld%(x,y)=feld%(x,y)+1
  wert%(zahl)=wert%(zahl)+1
  gx=305+(x*8)+(10*x):gy=39+(y*8)+(4*y)
```

```
c3   Gotoxy gx,gy:COLOR zahl+2,2:PRINT CHR$(feld%(x,y)+48)
z7   IF feld%(x,y)=init%(x,y) THEN i=0:GOSUB Ueberlauf
bp   IF spanz%-null=1 THEN GOSUB Endauswertung:RETURN
G3   zahl=zahl MOD spanz%+1
Bc   IF wert%(zahl)=-1 THEN zahl=zahl MOD spanz%+1
b4   GOSUB NeuerSpieler
jp   TIMER ON
iR1  ELSE
GJ2  SOUND 1000,4,255,0
5y0 END IF
W8  RETURN
EB  Ueberlauf:
Ag  IF x THEN
Gq2  IF feld%(x-1,y)<>-1 THEN
PJ4   IF feldfarbe%(x-1,y)<>zahl+2 AND feldfarbe%(x-1,y) T
      HEN
A36    wert%((feldfarbe%(x-1,y))-2)=wert%((feldfarbe%(x-1,y
       ))-2)-feld%(x-1,y)
t0     wert%(zahl)=wert%(zahl)+feld%(x-1,y)
D64   END IF
wF    feld%(x-1,y)=feld%(x-1,y)+1:feldfarbe%(x-1,y)=zahl+2
NJ    IF (feld%(x-1,y) MOD init%(x-1,y))=0 THEN i=i+1:stapel
      %(i,1)=x-1:stapel%(i,2)=y
nC    Gotoxy gx-18,gy:PRINT CHR$(feld%(x-1,y)+48)
HA2  END IF
IB0 END IF
LQ  IF x<14 THEN
Nv2  IF feld%(x+1,y)<>-1 THEN
S14   IF feldfarbe%(x+1,y)<>zahl+2 AND feldfarbe%(x+1,y) T
      HEN
va6    wert%((feldfarbe%(x+1,y))-2)=wert%((feldfarbe%(x+1,y
       ))-2)-feld%(x+1,y)
y3     wert%(zahl)=wert%(zahl)+feld%(x+1,y)
OH4   END IF
r4    feld%(x+1,y)=feld%(x+1,y)+1:feldfarbe%(x+1,y)=zahl+2
AQ    IF (feld%(x+1,y) MOD init%(x+1,y))=0 THEN i=i+1:stapel
      %(i,1)=x+1:stapel%(i,2)=y
aF    Gotoxy gx+18,gy:PRINT CHR$(feld%(x+1,y)+48)
SL2  END IF
TM0 END IF
b8  IF y THEN
Ku2  IF feld%(x,y-1)<>-1 THEN
BV4   IF feldfarbe%(x,y-1)<>zahl+2 AND feldfarbe%(x,y-1) T
      HEN
IS6    wert%((feldfarbe%(x,y-1))-2)=wert%((feldfarbe%(x,y-1
       ))-2)-feld%(x,y-1)
xs     wert%(zahl)=wert%(zahl)+feld%(x,y-1)
ZS4   END IF
ES    feld%(x,y-1)=feld%(x,y-1)+1:feldfarbe%(x,y-1)=zahl+2
d1    IF (feld%(x,y-1) MOD init%(x,y-1))=0 THEN i=i+1:stapel
      %(i,1)=x:stapel%(i,2)=y-1
Pb    Gotoxy gx,gy-12:PRINT CHR$(feld%(x,y-1)+48)
dW2  END IF
eX0 END IF
ms  IF y<14 THEN
Nv2  IF feld%(x,y+1)<>-1 THEN
6M4   IF feldfarbe%(x,y+1)<>zahl+2 AND feldfarbe%(x,y+1) T
      HEN
BF6    wert%((feldfarbe%(x,y+1))-2)=wert%((feldfarbe%(x,y+1
       ))-2)-feld%(x,y+1)
yr     wert%(zahl)=wert%(zahl)+feld%(x,y+1)
kd4   END IF
xF    feld%(x,y+1)=feld%(x,y+1)+1:feldfarbe%(x,y+1)=zahl+2
SR    IF (feld%(x,y+1) MOD init%(x,y+1))=0 THEN i=i+1:stapel
      %(i,1)=x:stapel%(i,2)=y+1
MU    Gotoxy gx,gy+12:PRINT CHR$(feld%(x,y+1)+48)
oh2  END IF
p10 END IF
1M  feld%(x,y)=feld%(x,y)-init%(x,y)
cH  Gotoxy gx,gy
J9  IF feld%(x,y)=0 THEN PRINT " ":feldfarbe%(x,y)=0 ELSE PRIN
    T CHR$(feld%(x,y)+48)
mh  j=0
Zd  FOR k=1 TO spanz%
Am2  IF wert%(k)=0 THEN null=null+1:wert%(k)=NOT wert%(k):IF
     spanz%-null<>1 THEN Meldung spielername$(k)+" ist ausge
     schieden",k+2 ELSE k=spanz%:j=-1
5A0 NEXT
D7  IF j THEN RETURN
WH  IF i THEN x=stapel%(i,1):y=stapel%(i,2):gx=305+(x*8)+(10*x
    ):gy=39+(y*8)+(4*y):i=i-1:GOTO Ueberlauf
P1  RETURN
6C  Computer:
to  MOUSE OFF:DIM kont(225,2):i=0
OJ  FOR  y=0 TO 14
vY2  FOR x=0 TO 14
S93   IF feldfarbe%(x,y)=zahl+2 AND feld%(x,y)=init%(x,y)-1 T
      HEN i=i+1:kont(i,1)=x:kont(i,2)=y
EJ2  NEXT
FK0 NEXT
LY  hprior=0:k=0
zC  WHILE hz>1 AND k<i
hQ2  prior=0:k=k+1:x=kont(k,1):y=kont(k,2)
YP   IF x>0 THEN
UA4   IF feldfarbe%(x-1,y)=zahl+1 THEN
jR6    prior=prior+1
AL     IF feld%(x-1,y)=init%(x-1,y)-1 THEN prior=prior+1
E74   END IF
F82  END IF
IN   IF x<14 THEN
WA4   IF feldfarbe%(x+1,y)=zahl+1 THEN
pX6    prior=prior+1
29     IF feld%(x+1,y)=init%(x+1,y)-1 THEN prior=prior+1
KD4   END IF
LE2  END IF
ph   IF y>0 THEN
044   IF feldfarbe%(x,y-1)=zahl+1 THEN
vd6    prior=prior+1
Vs     IF feld%(x,y-1)=init%(x,y-1)-1 THEN prior=prior+1
QJ4   END IF
RK2  END IF
Zf   IF y<14 THEN
M04   IF feldfarbe%(x,y+1)=zahl+1 THEN
1j6    prior=prior+1
Pi     IF feld%(x,y+1)=init%(x,y+1)-1 THEN prior=prior+1
WP4   END IF
XQ2  END IF
Yf   IF prior>hprior THEN j=k:hprior=prior
4s0 WEND
s5  IF wert%(2)>3 OR hprior<>0 THEN
m02  IF i>0 THEN
W04   IF hprior=0 THEN k=INT(RND*i)+1:x=kont(k,1):y=kont(k,2
      ) ELSE x=kont(j,1):y=kont(j,2)
dW2  END IF
eX0 END IF
LK  IF i=0 OR (wert%(2)<4 AND hprior=0) THEN
KB2  x=INT(RND*15):y=INT(RND*15)
1v   WHILE feldfarbe%(x,y)=zahl+1 OR feld%(x,y)=-1
MD   x=INT(RND*15):y=INT(RND*15)
E2   WEND
kd0 END IF
oU  ERASE kont:GOSUB Setzen2:MOUSE ON
Co  RETURN
1Y  ZeitAuswertung:
D7  IF flag%=24 OR flag%=15 THEN
Bk1  IF NOT komp OR (zahl=1 AND komp) THEN MOUSE STOP
Ir   sec%(zahl)=sec%(zahl)+1
cT   IF sec%(zahl)=60 THEN sec%(zahl)=0:min%(zahl)=min%(zahl)+
     1
rD   COLOR zahl+2,0: LOCATE 13+2*zahl,25
2X   IF min%(zahl)<10 THEN PRINT "0";
5G   PRINT RIGHT$(STR$(min%(zahl)),27-POS(0));".";
oU   IF sec%(zahl)<10 THEN PRINT "0";
uB   PRINT RIGHT$(STR$(sec%(zahl)),30-POS(0))
KF   hz=hz-1
Bb   IF hz<4 THEN SOUND 800,2,200,3
vs   IF hz=0 THEN
H43   zahl=zahl MOD spanz%+1
Cd    IF wert%(zahl)=-1 THEN zahl=zahl MOD spanz%+1
OU    SOUND 800,4,200,3:GOSUB NeuerSpieler
3w1  END IF
Jv   LOCATE 2,47:PRINT USING"##";hz
Ug   IF NOT komp OR (zahl=1 AND komp) THEN MOUSE ON
6z0 END IF
X9  RETURN
1Q  NeuerSpieler:
5K  COLOR zahl+2,0
8k  hz=zeit
6b  LOCATE 2,31:Printe "Denkzeit "+CHR$(252)+"brig: ":PRINT US
    ING"##";hz
```

Listing 2. Ein Spiel mit »allem Drum und Dran«: »Buh-Au-Construction-Set«. Bitte mit dem Checksummer (Seite 143) eingeben (Fortsetzung).

```
9n  LOCATE 10,2:Printe spielername$(zahl)+" ist am Zug."+SPACE
    $(20-LEN(spielername$(zahl)))
dF  RETURN
tp  Checker:
yp  IF zeiger THEN
fD2   GOSUB Wertung
hJ    RETURN
IBO END IF
9g  IF anzahl<4 THEN
I22   IF anzahl=0 THEN zeiger=-3 ELSE zeiger=-2
kI    GOSUB Wertung
mO    RETURN
NGO END IF
Up  IF einrfld THEN
M52   FOR i=1 TO einrfld
JK4     x=einr(i,0):y=einr(i,1):GOSUB Kaestchen
af2   NEXT
SLO END IF
ZS  ERASE init:DIM init%(14,14)
rg  gesamt=0:zeiger=0:i=1:j=anzahl:einrfld=0
Jp  FOR yt=0 TO 14
Eo2   FOR xt=0 TO 14
BX4     IF feld%(xt,yt)=0 THEN feld%(xt,yt)=1:GOSUB Checkbegin
        :yt=14:xt=14
hm2   NEXT
inO NEXT
H9  IF zeiger<>-4 THEN
kA2   IF j=0 THEN zeiger=1 ELSE zeiger=-1
cVO END IF
1Q  GOSUB Loeschen2:GOSUB Wertung
4g  RETURN
mk  Checkbegin:
1S  WHILE i
vr  i=i-1
9i  IF init%(xt,yt)=0 THEN
162   IF xt THEN
q94     IF feld%(xt-1,yt)>-1 THEN
cK6       feld%(xt-1,yt)=1:init%(xt,yt)=init%(xt,yt)+1
tU        IF init%(xt-1,yt)=0 THEN i=i+1:stapel%(i,1)=xt-1:sta
          pel%(i,2)=yt
ng4     END IF
oh2   END IF
NK    IF xt<14 THEN
q74     IF feld%(xt+1,yt)>-1 THEN
YE6       feld%(xt+1,yt)=1:init%(xt,yt)=init%(xt,yt)+1
nK        IF init%(xt+1,yt)=0 THEN i=i+1:stapel%(i,1)=xt+1:sta
          pel%(i,2)=yt
tm4     END IF
un2   END IF
z0    IF yt THEN
vM4     IF feld%(xt,yt-1)>-1 THEN
1v6       feld%(xt,yt-1)=1:init%(xt,yt)=init%(xt,yt)+1
SE        IF init%(xt,yt-1)=0 THEN i=i+1:stapel%(i,1)=xt:stape
          l%(i,2)=yt-1
zs4     END IF
Ot2   END IF
ec    IF yt<14 THEN
p04     IF feld%(xt,yt+1)>-1 THEN
1t6       feld%(xt,yt+1)=1:init%(xt,yt)=init%(xt,yt)+1
E6        IF init%(xt,yt+1)=0 THEN i=i+1:stapel%(i,1)=xt:stape
          l%(i,2)=yt+1
5y4     END IF
6z2   END IF
bB    IF init%(xt,yt)=1 THEN
SE4     einrfld=einrfld+1:einr(einrfld,0)=xt
Hg      einr(einrfld,1)=yt:zeiger=-4:x3=300+xt*18:x4=316+xt*18
MB      y3=yt*12:y4=10+yt*12:LINE(x3,y3)-(x4,y4),4,bf
B42   END IF
Rl    j=j-1:gesamt=gesamt+init%(xt,yt)
D60 END IF
rM  xt=stapel%(i,1):yt=stapel%(i,2)
kY  WEND
gI  RETURN
O4  Loeschen:
Gp  FOR y=0 TO 14
Bo2   FOR x=0 TO 14
re4     feld%(x,y)=-1
Fg      IF flag%=4 THEN GOSUB Kaestchen
Va2   NEXT
WbO NEXT
HL  IF flag%=4 THEN
Xf2   Neusetzen x1,y1,x2,y2,farbe%,wort$:anzahl=0:zeiger=0
QJO END IF
rT  RETURN
Tg  Fuellen:
RO  FOR y=0 TO 14
qV2   FOR x=0 TO 14:feld%(x,y)=0:GOSUB Kaestchen:NEXT
eJO NEXT
fu  Neusetzen x1,y1,x2,y2,farbe%,wort$:anzahl=225:zeiger=0
xZ  RETURN
9P  Loeschen2:
X6  FOR y=0 TO 14
AB2   FOR x=0 TO 14:IF feld%(x,y)=1 THEN feld%(x,y)=0
kp    NEXT
1qO NEXT
3f  RETURN
kl  Wertung:
oB  RESTORE Fehlermeldung
tz  j=zeiger
NH  IF zeiger=1 THEN j=0
QR  FOR i=1 TO j+5:READ Wertung$:NEXT
zN  Meldung Wertung$,4:Neusetzen x1,y1,x2,y2,farbe%,wort$
Am  RETURN
Zw  Fehlermeldung:
49  DATA Einerfeld vorhanden,Kein Spielfeld vorhanden,Zu klein
    es Spielfeld
D1  DATA Kein korrektes Spielfeld,Spielfeld ist zugelassen
nI  Anklicken:
Ve  IF POINT(mx,my)=0 OR POINT(mx,my)=1 THEN RETURN
Ua  xh=(mx-300)\18:yh=my\12:x=xh:y=yh
Ne  Marke:
i22   IF feld%(x,y)=-1 THEN
XO4     feld%(x,y)=0:anzahl=anzahl+1
ZI3   ELSE
ws4     feld%(x,y)=-1:anzahl=anzahl-1
wp2   END IF
z2    GOSUB Kaestchen
GI    WHILE x=xh AND y=yh AND FN Maus=0
xd4     mx=MOUSE(0):mx=MOUSE(5):my=MOUSE(6)
bA      xh=(mx-300)\18:yh=my\12
p2      IF yh>14 OR xh>14 OR yh<0 OR xh<0 THEN yh=y:xh=x
XL2   WEND
BT    x=xh:y=yh
YxO IF FN Maus=0 THEN GOTO Marke
a4  zeiger=0
W8  RETURN
Ab  Anklicken2:
p2  IF hwort$<>"" THEN
cV2   LINE (x7+1,y7+1)-(x8-1,y8-1),0,bf
r9    COLOR 2,0:LOCATE 2,3+10*(x7\80):Printe hwort$
B40 END IF
Ab  LINE (x1,y1)-(x2,y2),2,bf:COLOR 0,2
IS  LOCATE 2,3+10*(x1\80):Printe wort$:COLOR 3,0
Gr  x7=x1:y7=y1:x8=x2:y8=y2:hwort$=wort$:zeit=5*(x1\80)
fH  RETURN
Dk  Sicher:
RL  Wertung$="Sind Sie 100%ig sicher ??":GOSUB Requester
iK  RETURN
YN  Sicher2:
me  Wertung$="File existiert !! SAVE ??":GOSUB Requester
1N  RETURN
S3  Requester:
Zm  flag%=-3
ZE  WhileMousePressed
CQ  WINDOW 3,,(196,58)-(428,101),16,1:LINE (0,0)-(232,52),6,b
tG  RESTORE Daten7:Zeichnen 2
t2  COLOR 4,0:LOCATE 2,3:Printe Wertung$:ja=1
1t  WHILE ja=1
tH2   call WaitForClick:mx=MOUSE(0):GOSUB MouseTest
znO WEND
qX  WINDOW CLOSE 3
wY  RETURN
oT  ja:
AS  IF flag%=-2 THEN ja=-1 ELSE ja=0
zb  RETURN
nh  Out:
5B  Neusetzen x1,y1,x2,y2,farbe%,wort$
uM  num=flag%-6:c$=file$(0)
RW  GOSUB LiesDiskName:WBenchToBack
pH  IF c$<>file$(0) THEN disk$=drive$:GOSUB LeseDir
rC  efile=1:edir=0:WINDOW 3,,(165,14)-(475,218),0,1
po  COLOR 6,0:Gotoxy 4,6:POKE WINDOW(8)+56,2:Printe file$(0)
X4  POKE WINDOW(8)+56,0:Gotoxy 4,18:Printe "Verzeichnis: "
DX  Gotoxy 4,30:Printe disk$:LINE (4,32)-(438,32),6
mt  FileKasten 0,farbe%:GOSUB DirAnzeige:FileKasten 72,farbe%
Sn  GOSUB FileAnzeige
```

```
CC  a$=RIGHT$(wort$,LEN(wort$)-INSTR(wort$," "))
uX  Gotoxy 4,190:Printe "Zu "+a$+"des File:"
zc  Gotoxy 4,202:Printe namen$(num)
6B  LINE (4,205)-(186,205):RESTORE Daten3:Zeichnen 4+df1%:k=0
K6  WHILE k<7 OR k>8
dQ2   call WaitForClick:mx=MOUSE(0):mx=MOUSE(1)
4u    my=MOUSE(2):k=0:x1=0
aR    RESTORE Daten3
CC    WHILE x1<>-1 AND k=0
aF4     READ x1,y1,x2,y2,flag%,i,wort$
E6      IF x1<>-1 AND mx>x1 AND my>y1 AND mx<x2 AND my<y2 THEN k=flag%
RF2   WEND
30    IF k>6 AND k<11+df1% THEN farbe%=i:Zeichnen2 x1,y1,x2,y2,farbe%,wort$
D3    IF x1<>-1 THEN
rs4     ON k GOSUB SelDir,DirUp,DirDown,SelFile,FileUp,FileDown,Ok,Abbruch,ChDisk,ChDisk,SaveFile
Ot2   END IF
WKO WEND
S4  RETURN
eP  ChDisk:
ZM  IF k=10 AND df1% THEN RETURN
Zf  Neusetzen x1,y1,x2,y2,farbe%,wort$
nk  IF (k=9 AND drive$="df0:") OR (k=10 AND drive$="df1:") THEN
eK2   c$=file$(0)
w1    GOSUB LiesDiskName:WBenchToBack
Pv    IF c$=file$(0) AND disk$=drive$ THEN RETURN
t9    disk$=drive$
qZ1  ELSE
KU2   MID$(drive$,3,1)=CHR$(48+ABS(VAL(MID$(drive$,3,1))-1))
jp    c$=disk$:disk$=drive$:x=0:GOSUB LiesDiskName:WBenchToBack
5n    IF x=-1 THEN disk$=c$:drive$="df0:":RETURN
F80 END IF
E1  COLOR 6,0:Gotoxy 4,6:POKE WINDOW(8)+56,2
rQ  Printe file$(0)+SPACE$(50):POKE WINDOW(8)+56,0
hl  GOSUB NeuesDir
jL  RETURN
sh  SelFile:
J5  y=(my-113)\8:x=y
MR  IF y>fanz%-1 THEN RETURN
Y6  FileLoop:
02  IF y>fanz%-1 THEN y=fanz%-1
Cb  IF x<>y THEN
Ha2   COLOR 6,0:LOCATE 15+x,2
OM    Printe file$(efile+x)+SPACE$(22-LEN(file$(efile+x))):x=y
SLO END IF
j6  COLOR 0,6
GE  LOCATE 15+y,2:Printe file$(efile+y)+SPACE$(22-LEN(file$(efile+y)))
xV  mx=MOUSE(0):mx=MOUSE(1):my=MOUSE(2)
oL  IF mx>x1 AND mx<x2 AND my>y1 AND my<y2 THEN y=(my-113)\8
5X  IF FN Maus=0 THEN FileLoop
Pi  COLOR 6,0:LOCATE 15+x,2
gd  Printe file$(efile+x)+SPACE$(22-LEN(file$(efile+x)))
91  namen$(num)=file$(efile+x)
oF  Gotoxy 4,202:Printe namen$(num)+SPACE$(22-LEN(namen$(num)))
2e  RETURN
SE  SelDir:
A7  IF RIGHT$(disk$,1)=":" THEN i=1:j=5 ELSE i=0:j=6
EE  y=(my-41)\8+i:x=y
gk  IF y>danz THEN RETURN
2i  DirLoop:
AW  IF edir+y>danz THEN y=danz-edir
Wv  IF x<>y THEN
bH2   COLOR 6,0:LOCATE j+x,2
qY    Printe dir$(edir+x)+SPACE$(22-LEN(dir$(edir+x))):x=y
mf0 END IF
3Q  COLOR 0,6
hb  LOCATE j+y,2:Printe dir$(edir+y)+SPACE$(22-LEN(dir$(edir+y)))
Hp  mx=MOUSE(0):mx=MOUSE(1):my=MOUSE(2)
Kr  IF mx>x1 AND mx<x2 AND my>y1 AND my<y2 THEN y=(my-41)\8+i
OK  IF FN Maus=0 THEN DirLoop
jP  COLOR 6,0:LOCATE j+x,2
7B  Printe dir$(edir+x)+SPACE$(22-LEN(dir$(edir+x)))
Nh  IF edir+x=0 THEN
1A2   WHILE RIGHT$(disk$,1)<>"/" AND RIGHT$(disk$,1)<>":"
1j4     disk$=LEFT$(disk$,LEN(disk$)-1)
SG2   WEND
cz    IF LEN(disk$)<>4 THEN disk$=LEFT$(disk$,LEN(disk$)-1)
eN1  ELSE
tN2   IF RIGHT$(disk$,1)<>":" THEN disk$=disk$+"/"
SF    disk$=disk$+dir$(edir+x)
2v0 END IF
ua  c$=file$(0):GOSUB LiesDiskName:WBenchToBack
pM  IF c$<>file$(0) THEN
qm2   Meldung "Falsche Disk in Laufwerk "+MID$(drive$,3,1),4
VV    file$(0)=c$:GOSUB Abbruch:k=8
X9    RETURN
810 END IF
Cg  NeuesDir:
oN  Gotoxy 4,30:Printe disk$+SPACE$(40)
h6  FOR i=1 TO 8
552   LOCATE 14+i,2:Printe SPACE$(22):LOCATE 5+i,2:Printe SPACE$(22)
MRO NEXT
fu  FOR x=1 TO 3:namen$(x)="":NEXT
St  Gotoxy 4,202:Printe namen$(num)+SPACE$(22-LEN(namen$(num)))
kE  GOSUB LeseDir:WBenchToBack
x5  edir=0:efile=1
7A  GOSUB DirAnzeige:GOSUB FileAnzeige
jL  RETURN
YP  SaveFile:
NK  IF num<>2 THEN RETURN
yn  Gotoxy 4,202:Eingabe 22,6,namen$(num)
nP  RETURN
Ft  DirDown:
71  Invertieren
4e  DD:
Z22   IF danz-edir>8 THEN edir=edir+1:GOSUB DirAnzeige
2KO IF FN Maus=0 THEN DD
B5  Invertieren
uW  RETURN
Pb  DirUp:
E8  Invertieren
Or  DU:
k82   IF edir>0 THEN edir=edir-1:GOSUB DirAnzeige
nz0 IF FN Maus=0 THEN DU
IC  Invertieren
1d  RETURN
HR  FileDown:
LF  Invertieren
My  FD:
IU2   IF fanz%-efile>7 THEN efile=efile+1:GOSUB FileAnzeige
QkO IF FN Maus=0 THEN FD
PJ  Invertieren
8k  RETURN
zP  FileUp:
SM  Invertieren
IB  FU:
TT2   IF efile>1 THEN efile=efile-1:GOSUB FileAnzeige
BPO IF FN Maus=0 THEN FU
WQ  Invertieren
Fr  RETURN
Ij  DirAnzeige:
W3  IF RIGHT$(disk$,1)=":" THEN k1=1 ELSE k1=0
No  IF danz<8 THEN h=danz ELSE h=8
r2  LOCATE 5,2
yL  COLOR 6,0
Oh  FOR i=k1+edir TO edir+h
292   LOCATE CSRLIN+1,2:Printe dir$(i)+SPACE$(22-LEN(dir$(i)))
6BO NEXT
OO  RETURN
2z  FileAnzeige:
sC  IF fanz%<8 THEN h=fanz% ELSE h=8
NI  LOCATE 14,2
6T  COLOR 6,0
BM  FOR i=efile TO h-1+efile
rT2   LOCATE CSRLIN+1,2:Printe file$(i)+SPACE$(22-LEN(file$(i)))
EJO NEXT
W8  RETURN
OO  Ok:
bh  Neusetzen x1,y1,x2,y2,farbe%,wort$:c$=file$(0)
```

Listing 2. Ein Spiel mit »allem Drum und Dran«: »Buh-Au-Construction-Set«. Bitte mit dem Checksummer (Seite 143) eingeben (Fortsetzung).


```
695 x2  GOSUB LiesDiskName:WBenchToBack
696 vS  IF c$<>file$(0) THEN
697 ws2   Meldung "Falsche Disk in Laufwerk "+MID$(drive$,3,1),4
698 Kq    file$(0)=c$:GOSUB Abbruch
699 dF    RETURN
700 E70 END IF
701 m5  IF namen$(num)="" THEN
702 DY2   Meldung "Kein Filename",4:GOSUB Abbruch
703 hJ    RETURN
704 IBO END IF
705 WL  ON num GOSUB Laden,Speichern,DelFile
706 aa  flag%=3
707 1N  RETURN
708 Wm  Speichern:
709 ir  x=0
710 UD  FOR i=1 TO fanz%
711 oL2   IF UCASE$(namen$(num))=UCASE$(file$(i)) THEN x=-1:i=fanz
        %
712 ZeO NEXT
713 90  IF x THEN GOSUB Sicher2:IF NOT ja THEN RETURN
714 Bq  OPEN disk$+namen$(num)+".fld" FOR OUTPUT AS #1
715 RO2   FOR y=0 TO 14
716 Hn4     FOR x=0 TO 14:WRITE#1,feld%(x,y),init%(x,y):NEXT
717 ej2   NEXT
718 Bc    WRITE#1,zeiger,anzahl,gesamt
719 VIO CLOSE#1
720 fr  KILL disk$+namen$(num)+".fld.info"
721 WI  WBenchToBack:WINDOW CLOSE 3:h=0
722 gP  FOR i=1 TO fanz%
723 102   IF UCASE$(namen$(num))=UCASE$(file$(i)) THEN h=-1
724 1qO NEXT
725 Jb  IF h THEN RETURN
726 ki  fanz%=fanz%+1:file$(fanz%)=namen$(num)
727 OD  GOSUB FileSort
728 61  RETURN
729 f1  Laden:
730 LV  OPEN disk$+namen$(num)+".fld" FOR INPUT AS #1
731 412   WINDOW CLOSE 3
732 1H    FOR y=0 TO 14
733 dG4     FOR x=0 TO 14
734 a16       INPUT #1,feld%(x,y),init%(x,y):GOSUB Kaestchen
735 w14     NEXT
736 x22   NEXT
737 p8    INPUT#1,zeiger,anzahl,gesamt
738 obO CLOSE#1
739 Ht  RETURN
740 wW  DelFile:
741 op  KILL disk$+namen$(num)+".fld"
742 OJ  FOR i=1 TO fanz%
743 3R2   IF file$(i)=namen$(num) THEN h=i
744 5AO NEXT
745 uS  FOR i=h TO fanz%:SWAP file$(i+1),file$(i):NEXT
746 hq  IF namen$(num)=namen$(1) THEN namen$(1)=""
747 oE  namen$(num)="":fanz%=fanz%-1:WINDOW CLOSE 3
748 Q2  RETURN
749 lv  Abbruch:
750 N4  WINDOW CLOSE 3
751 SA  IF flag%=14 THEN flag%=22 ELSE flag%=3
752 U6  RETURN
753 qr  Quit:
754 ag  Neusetzen x1,y1,x2,y2,farbe%,wort$
755 J3  GOSUB Sicher
756 aW  IF NOT ja THEN flag%=1:RETURN
757 Hh  SCREEN CLOSE 1:WINDOW CLOSE 2:FreeMem info&,252&
758 yt  END
759 u4  Spielerauswahl:
760 gm  Neusetzen x1,y1,x2,y2,farbe%,wort$
761 bP  WINDOW 3,,(200,64)-(440,160),16,1
762 6z  RESTORE Daten5
763 AZ  IF spanz%<>y2\50 THEN
764 b42   FOR i=1 TO 3:spielername$(i)="":NEXT
765 HAO END IF
766 uT  spanz%=y2\50:Zeichnen spanz%+1:COLOR 6,0
767 KU  FOR y=1 TO spanz%
768 yq2   LOCATE 2*y+INT(y/1.5),2
769 5A    IF spanz%=1 THEN PRINT"Name   :"; ELSE PRINT"Name";y":";
770 Mm    Printe spielername$(y)
771 WbO NEXT
772 JF  k=0
773 s4  FOR i=1 TO spanz%
774 9t2   y=2*i+INT(i/1.5):LOCATE y,10:Eingabe 19,6,spielername$(i
      )
775 9h    IF MOUSE(0) THEN i=spanz%:k=-1
776 bgO NEXT
777 1C  IF k THEN GOSUB MouseTest
778 uW  RETURN
779 Vs  SpielerEingabe:
780 5G  j=y2\18
781 WY  IF j>spanz% THEN RETURN
782 R9  y=2*j+INT(j/1.5):LOCATE y,10:Eingabe 19,6,spielername$(j)
783 Mi  IF MOUSE(0) THEN GOSUB MouseTest
784 Oc  RETURN
785 N5  Kaestchen:
786 1B  x3=300+x*18:x4=x3+16:y3=y*12:y4=y3+10
787 Tn  IF feld%(x,y)=-1 THEN
788 vv2   LINE(x3,y3)-(x4,y4),6,bf
789 K31 ELSE
790 p12   LINE(x3,y3)-(x4,y4),2,bf
791 haO END IF
792 8k  RETURN
793 tM  Endauswertung:
794 Vg  MOUSE OFF:CLS
795 eo  IF komp THEN komp=0:spanz%=1
796 xO  Meldung spielername$(zahl)+" hat gewonnen !!!",zahl+2
797 4D  COLOR 6,0:CLS:LOCATE 5,2:Printe "Die Punkte:"
798 HT  FOR i=1 TO spanz%
799 422   j=(min%(i)*60+sec%(i))
800 eC    IF wert%(i)>0 THEN punkte%(i)=INT((wert%(i)/gesamt*anza
      hl*(30-(j/100)))*1.536421) ELSE punkte%(i)=j+5
801 Op    IF punkte%(i)<=0 THEN punkte%(i)=5
802 1J    COLOR i+2,0:LOCATE 5+2*i,4
803 eu    Printe spielername$(i)+SPACE$(20-LEN(spielername$(i)))+"
      :"
804 6L    PRINT USING"####";punkte%(i)
805 490 NEXT
806 uF  COLOR 4,0
807 hG  LOCATE 17,27:Printe "Linke Maustaste dr"+CHR$(252)+"cken !
      !"
808 Dx  CALL WaitForClick:CLS:x=0
809 Se  FOR i=1 TO spanz%
810 Lu  IF punkte%(i)>hsc%(13) THEN
811 GY2   hsc%(13)=punkte%(i):hsc$(13)=spielername$(i):j=13:x=-1
812 p7    WHILE hsc%(j)>hsc%(j-1) AND j>1
813 sa4     SWAP hsc%(j),hsc%(j-1):SWAP hsc$(j),hsc$(j-1):j=j-1
814 ZN2   WEND
815 kTO ELSE
816 fw2   IF punkte%(i)<lsc%(13) THEN
817 c24     lsc%(13)=punkte%(i):lsc$(13)=spielername$(i):j=13:x=-1
818 Vq      WHILE lsc%(j)<lsc%(j-1)
819 wu6       SWAP lsc%(j),lsc%(j-1):SWAP lsc$(j),lsc$(j-1):j=j-1
820 fT4     WEND
821 B42   END IF
822 C50 END IF
823 MR  NEXT
824 nT  IF x=-1 THEN GOSUB ScSpeichern
825 Fy  Farbenloeschen
826 bo  COLOR 3,0:LOCATE 2,8:Printe "13 der Besten..."
827 P3  LOCATE 2,47:Printe "und Schlechtesten."
828 XJ  LINE (0,16)-(640,18),,bf:LINE (320,0)-(321,198),,bf
829 iJ  FOR i=1 TO 13
830 xP2   j=25+i*13:COLOR 2,0:Gotoxy 8,j:PRINT USING"##";i;
831 xC    PRINT ".":COLOR 5,0:Gotoxy 64,j:Printe hsc$(i)
832 CG    Gotoxy 232,j:PRINT USING"#####";hsc%(i);
833 fc    COLOR 6,0:Gotoxy 376,j:Printe lsc$(i)
834 L9    Gotoxy 544,j:PRINT USING"#####";lsc%(i)
835 zD    j=29+i*13:LINE(0,j)-(640,j),7
836 ZeO NEXT
837 ik  COLOR 4,0:LOCATE 28,27
838 pR  Printe "Linke Maustaste dr"+CHR$(252)+"cken !!"
839 ms  call Farbensetzen :call WaitForClick :CLS
840 Tc  GOSUB NeustartInitialisierungsRoutine:GOSUB Aufbau1
841 vX  RETURN
842 zQ  NeustartInitialisierungsRoutine:
843 zg  ERASE wert%,feldfarbe%:DIM wert%(3),feldfarbe%(14,14)
844 W5  FOR y=0 TO 14
845 R42   FOR x=0 TO 14
846 DO4     IF feld%(x,y)>0 THEN feld%(x,y)=0
847 kp2   NEXT
848 1qO NEXT
849 ef  null=0:flag%=1
850 vF  FOR i=1 TO 3
851 DA2   IF sbc%(i)=-1 THEN spielername$(i)=""
852 puO NEXT
853 7j  RETURN
```

```
LeseDir:
FOR i=0 TO fanz%:file$(i)="":NEXT
FOR i=0 TO danz:dir$(i)="":NEXT
GOSUB LiesDiskName
fanz%=0:danz=0:dir$(0)="/ Parent":ex&=Lock&(SADD(a$),-2)
dummy&=Examine&(ex&,info&):dummy&=ExNext&(ex&,info&)
WHILE dummy&<>0
  fanz%=fanz%+1:file$(fanz%)="":i=0
  GOSUB LiesName
  IF j=2 THEN danz=danz+1:dir$(danz)=file$(fanz%)
  IF RIGHT$(file$(fanz%),4)<>".fld" THEN
    file$(fanz%)="":fanz%=fanz%-1
   ELSE
    file$(fanz%)=LEFT$(file$(fanz%),LEN(file$(fanz%))-4)
  END IF
  dummy&=ExNext&(ex&,info&)
WEND
FOR i=1 TO danz
  FOR j=1+i TO danz
    IF UCASE$(dir$(j))<UCASE$(dir$(i)) THEN SWAP dir$(j),
    dir$(i)
  NEXT
NEXT
FileSort:
FOR i=1 TO fanz%
  FOR j=1+i TO fanz%
    IF UCASE$(file$(j))<UCASE$(file$(i)) THEN SWAP file$(
    j),file$(i)
  NEXT
NEXT
ex&=UnLock&(ex&)
RETURN
LiesDiskName:
a$=drive$+CHR$(0):ex&=Lock&(SADD(a$),-2)
IF ex&=0 THEN x=-1
dummy&=Examine&(ex&,info&):i=0:file$(0)="":h=fanz%:fanz%=0
GOSUB LiesName:fanz%=h:a$=disk$+CHR$(0):ex&=UnLock&(ex&)
RETURN
LiesName:
j=PEEK(info&+7):b$=CHR$(PEEK(info&+i+8))
IF ASC(b$)=0 THEN RETURN
file$(fanz%)=file$(fanz%)+b$:i=i+1
GOTO LiesName
ScLaden:
OPEN UrDisk$+":SCORES.bcs" FOR INPUT AS 1
FOR i=1 TO 13:INPUT#1,hsc$(i),hsc%(i),lsc$(i),lsc%(i):NEX
T
CLOSE 1
RETURN
ScSpeichern:
OPEN UrDisk$+":SCORES.bcs" FOR OUTPUT AS 1
FOR i=1 TO 13:WRITE #1,hsc$(i),hsc%(i),lsc$(i),lsc%(i):NE
XT
CLOSE 1:KILL UrDisk$+":SCORES.bcs.info"
WBenchToBack
RETURN
HighScores:
IF NOT ScoreFehler% THEN GOSUB ScLaden:RETURN
RESTORE HighDaten
FOR i=1 TO 13:READ hsc$(i),hsc%(i),lsc$(i),lsc%(i):NEXT
GOSUB ScSpeichern
RETURN
HighDaten:
DATA K. Schreck,4000,Murphy,10,M.Boner,3000,Murphy,20,AMIG
A,1000
DATA Murphy,30,AMIGA,800,Murphy,40,AMIGA,700,Murphy,50,AMI
GA,650
DATA Murphy,70,AMIGA,600,Murphy,90,AMIGA,550,Murphy,110,AM
IGA,500
DATA Murphy,130,AMIGA,450,Murphy,150,AMIGA,400,Murphy,170,
AMIGA,300
DATA Murphy,190,AMIGA,200,Murphy,200
SUB Eingabe(max%,farbe%,eing$) STATIC
COLOR farbe%,0:Printe eing$:COLOR 0,farbe%
PRINT" ";:COLOR farbe%,0:a$=""
WHILE INKEY$<>"":WEND
WHILE MOUSE(0):WEND
WHILE a$<>CHR$(13) AND MOUSE(0)=0
  a$="":a$=INKEY$
  IF a$<>"" THEN
    IF ASC(a$)>31 AND LEN(eing$)<max% THEN
      eing$=eing$+a$:PRINT CHR$(8);a$;
      COLOR 0,farbe%:PRINT " ";:COLOR farbe%,0
    END IF
    IF a$=CHR$(8) AND LEN(eing$)>0 THEN
      PRINT CHR$(8);CHR$(8);:COLOR 0,farbe%
      PRINT " ";:COLOR farbe%,0:PRINT " ";CHR$(8);
      eing$=LEFT$(eing$,LEN(eing$)-1)
    END IF
  END IF
WEND
PRINT CHR$(8);
END SUB
SUB Zeichnen(max%) STATIC
FOR i=1 TO max%
  READ x1,y1,x2,y2,flag%,farbe%,wort$
  LINE (x1,y1)-(x2,y2),farbe%,b:COLOR farbe%,0
  x=x1+(x2-x1-LEN(wort$)*8)\2:y=y1+(y2-y1+4)\2
  Gotoxy x,y:Printe wort$
NEXT
END SUB
SUB Zeichnen2(x1,y1,x2,y2,farbe%,wort$) STATIC
LINE (x1,y1)-(x2,y2),farbe%,bf:COLOR 0,farbe%
x=x1+(x2-x1-LEN(wort$)*8)\2:y=y1+(y2-y1+4)\2
Gotoxy x,y:Printe wort$:COLOR farbe%,0
WhileMousePressed
END SUB
SUB Printe (a$) STATIC
Text WINDOW(8),SADD(a$),LEN(a$)
END SUB
SUB Meldung(Wertung$,exfarbe%) STATIC
x=((640-((LEN(Wertung$)+2)*8))\2)
WINDOW 3,,(x,80)-(x+(LEN(Wertung$)*8)+16,96),16,1
WINDOW OUTPUT 3
PAINT (0,0),exfarbe%:COLOR 0,exfarbe%:PRINT
Printe " "+Wertung$+" "
IF exfarbe%=2 THEN
  j!=TIMER
  WHILE TIMER<j!+1:WEND
 ELSE
  call WaitForClick
END IF
WINDOW CLOSE 3
END SUB
SUB Neusetzen(x1,y1,x2,y2,farbe%,wort$) STATIC
LINE (x1+1,y1+1)-(x2-1,y2-1),0,bf:COLOR farbe%,0
x=x1+(x2-x1-LEN(wort$)*8)\2:y=y1+(y2-y1+4)\2
Gotoxy x,y:Printe wort$
END SUB
SUB FileKasten(y,farbe%) STATIC
LINE (4,38+y)-(186,105+y),farbe%,b
LINE (186,38+y)-(210,105+y),farbe%,b
LINE (186,52+y)-(210,91+y),farbe%,b
AREA (198,42+y):AREA (192,48+y):AREA (204,48+y):AREAFILL
AREA (198,101+y):AREA (192,95+y):AREA (204,95+y):AREAFILL
END SUB
SUB Gotoxy(x,y) STATIC
Move WINDOW(8),x,y
END SUB
SUB WaitForClick STATIC
WHILE FN Maus:WEND
END SUB
SUB WhileMousePressed STATIC
WHILE FN Maus=0:WEND
END SUB
SUB Invertieren STATIC
SHARED x1,y1,x2,y2
SetDrMd WINDOW(8),2:LINE (x1,y1)-(x2,y2),6,bf
SetDrMd WINDOW(8),1
END SUB
SUB Farbensetzen STATIC
PALETTE 0,0,0,0:PALETTE 1,0,0,0:PALETTE 2,.4,.6,1
PALETTE 3,.33,.87,0:PALETTE 4,.93,.2,0:PALETTE 5,1,1,.13
PALETTE 6,.73,.73,.73:PALETTE 7,.8,.321,.95
END SUB
SUB Farbenloeschen STATIC
FOR x=0 TO 7:PALETTE x,0,0,0:NEXT
END SUB
(C) 1989 M&T
```

Listing 2. Ein Spiel mit »allem Drum und Dran«: »Buh-Au-Construction-Set«. Bitte mit dem Checksummer (Seite 143) eingeben (Schluß).

BÖRSENFIEBER AM AMIGA

Die Börse — dieser Begriff setzt Assoziationen frei. Der eine denkt an Spekulationsgewinne und Husarenstreiche, der andere an Crash und Verluste. Erleben Sie die Faszination der Börse »live« auf Ihrem Amiga — und das ohne finanzielles Risiko.

Kaufen und Verkaufen, durch geschicktes Manipulieren den Gegner übernahmereif machen. Dieses Thema stammt nicht aus dem Wallstreet-Journal, sondern von einer turbulenten Börsensimulation, die so manchen eingefleischten Broker ins Schwitzen bringen kann. »Broker« (Listing 1) ist ein Spiel für zwei bis vier Teilnehmer. Durch geschickte Aktionen versucht jeder, den Kurs der eigenen Aktien möglichst hoch zu treiben, und den der Mitspieler so tief wie möglich zu drücken.

Wer zuerst eine vorher vereinbarte Summe erreicht, ist Sieger und kann sich beinahe schon als Börsen-Profi fühlen. Sie benötigen allerdings sehr viel planerisches Geschick und ein wenig Glück, um gegen die Konkurrenz zu bestehen.

Alle wichtigen Kommandos werden mit der rechten Maustaste über Pull-Down-Menüs oder wahlweise über Tastenkombinationen (Shortcuts) aufgerufen, die man sich mit dem Menüpunkt **Help** (Bild 1) ansehen kann. Um ein Spiel zu beginnen, wählen Sie bitte **Neues Spiel** an. Nach Eingabe aller Spielernamen werden Sie aufgefordert, das Startkapital und den Betrag festzusetzen, bei dessen Erreichen das Spiel beendet wird. Der Mindestbetrag, der vom Sieger zu erreichen ist, muß mindestens zehnmal so groß sein wie das Startkapital.

```
BROKER - Das Börsenspiel von Sebastian Peetz

 Menüpunkt      Taste                Erklärung                               DM|HP
-----------------------------------------------------------------------------------
Laden           ALT l    Ein Spiel wird von Disk geladen                        |
Speichern       ALT s    Ein Spiel wird abgespeichert                           |
Neues Spiel     ALT n    Es wird ein neues Spiel begonnen                       |
Quit            ALT q    Spielende ohne die Spielstände abzuspeichern           |
Börsenaktion      a      Der Spieler kann aktiv die Börsenkurse               20|1-3
                         durch geschicktes Handeln beeinflussen                 |
Börsenkurse       b      Es werden die aktuellen Börsenkurse angezeigt        - | -
Charts            c      Aktienkurse können graphisch dargestellt werden     60| -
Aktienkauf        k      Aktien können zum Tageskurs gekauft werden          5%| 2
Aktienverk.       v      Aktien können zum Tageskurs verkauft werden         5%| 2
Kapitalliste      l      Eigenes Kapital (Aktienkapital und bar)              - | -
Kapitalvergl.     r      Kapital aller Spieler                               30| -
Aktienliste       i      Auflistung der eigenen Aktien (Kurse,Stück)          - | -
Aktienvergl.      g      Aktien aller Spieler (Stück)                        40| -
Nächster          n      Der nächste Spieler ist an der Reihe                   |
Uebersicht        u      Alle wichtigen Daten zum Spiel im überblick            |
Info            Help     Diese Seite                                            |

Bitte Taste drücken
```

Bild 1. Diese Übersicht erreichen Sie mit der »Help«-Taste oder mit Help aus dem Info-Menü

Broker's Basar

In jeder neuen Runde haben Sie fünf Handlungspunkte zur Verfügung, die Sie gegen Aktionen an der Börse tauschen können. Je nach Art der Aktion werden unterschiedliche Punktzahlen abgezogen. So kostet jeder Aktienkauf und -verkauf jeweils zwei Punkte, Manipulationen der Kurse (siehe unten) zwischen einem und drei Punkten. Daneben können Sie verschiedene Dienstleistungen in Anspruch nehmen, wobei die meisten bares Geld kosten. So können Sie sich für 60 Mark die Kursentwicklung ausgewählter Aktien ansehen oder für 30 Mark einen Vergleich des Kapitals aller Spieler ausgeben lassen. 40 Mark müssen Sie investieren, wenn Sie Informationen über die Aktien Ihrer Gegner wollen.

Sie sollten dabei bedenken: Jede Information schmälert zwar Ihr Bankkonto, aber eine Fehlinvestition kann weit wertvollere Handlungspunkte kosten. Sie können mit dem Menüpunkt »Börsenaktion« Aktienkurse teilweise gezielt beeinflussen. Voraussetzung hierfür ist, daß Sie die Besitzverhältnisse aller Spieler genau kennen: Wie viele Brauereiaktien hatte doch gleich die Konkurrenz, waren Versicherungsaktien nicht eben billiger geworden?

Wenn Sie keine Handlungspunkte mehr haben, können Sie nur noch Informationen einholen oder an den nächsten Spieler abgeben (mit dem Menüpunkt »Nächster«). Bei jedem Wechsel werden die Schulden, die sich im Verlauf der aktuellen Runde angesammelt haben, mit dem Barkapital verrechnet.

Das Spiel ist zu Ende, wenn das Nettokapital eines Spielers (Bargeld plus in Aktien angelegtes Kapital abzüglich Schulden) unter 100 Mark fällt oder ein Broker den festgelegten Spielendbetrag erreicht. Sind dabei noch Schulden vorhanden, so werden erst diese verrechnet. Sind die Schulden eines Spielers größer als die Hälfte des Startkapitals, wird das Spiel ebenfalls beendet.

Spekulieren Sie mit

Kapital erwirtschaftet der Broker, indem er Aktien zu einem möglichst billigen Kurs erwirbt (Bild 2) und diese zu einem hohen Preis wieder verkauft. Bei Kauf und Verkauf werden jeweils 5 Prozent des Umsatzes als Maklergebühren vom Bargeld abgezogen und zwei Handlungspunkte verbraucht. Damit der Spieler nicht den Überblick verliert, zeigt der Computer beim Verkauf der Aktien den Ankaufskurs und den entsprechenden Gewinn oder Verlust an.

```
Aktienkauf

Name: Boesky      Barkapital: 3704 DM

Welche Aktie möchten Sie kaufen?          (0) Keine
---------------------------------------------------
  Nr.        AG              Vorrätig       Kurs

 ( 1 )      Bauwerte           1732          167
 ( 2 )      Textil             1492          140
 ( 3 )      Brauerei           1488          278
 ( 4 )      Kosmetik           1250          282
 ( 5 )      Elektrizität       1250          428
 ( 6 )      Chemie             1000          428
 ( 7 )      Pharmazie          1000          604
 ( 8 )      Maschinenbau        750          574
 ( 9 )      Kaufhaus            750          728
 ( 10 )     Computer            500          760
 ( 11 )     Automobil           500          878
 ( 12 )     Banken              250          878
 ( 13 )     Versicherung        250          700
```

Bild 2. Das Schaufenster für den Aktieneinkauf

Beim Menüpunkt Börsenaktion geht es richtig los. Hier spielen sich alle Kursveränderungen ab. Allerdings haben Sie nicht unbegrenzt Einfluß auf die Kurse. Beim Aufruf einer Börsenaktion bestimmt der Zufall, welche Manipulationen ermöglicht werden. Die ganze Handlungspalette zeigt sich bei »Manual Activity II« (Bild 3). Hier können Sie wählen, welche Manipulation ausgeführt werden soll. Dazu bekommen Sie noch eine kurze Erklärung zur Wirkung der einzelnen Punkte.

Broker

Broker sind Leute, deren Job es ist, an der Börse mit Aktien und Wertpapieren zu handeln. Dabei werden Kursdifferenzen ausgenutzt, um Gewinne zu erzielen. Mit verschiedenen Techniken ist es möglich, Kurse zu beeinflussen. So steigt der Kurs von Aktien häufig auf das Gerücht hin, die Firma würde von einer anderen übernommen. Solche Tendenzen geschickt auszunutzen, ist Aufgabe eines guten Brokers.

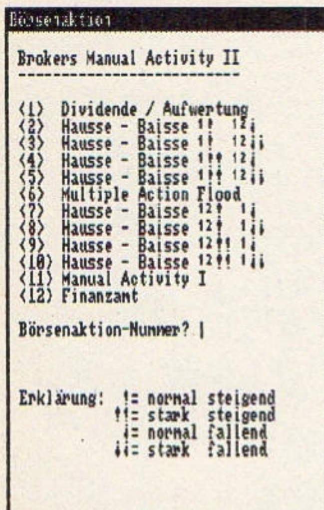


Bild 3. »Manual Activity II« zeigt Ihnen die ganze Trickkiste der Manipulation

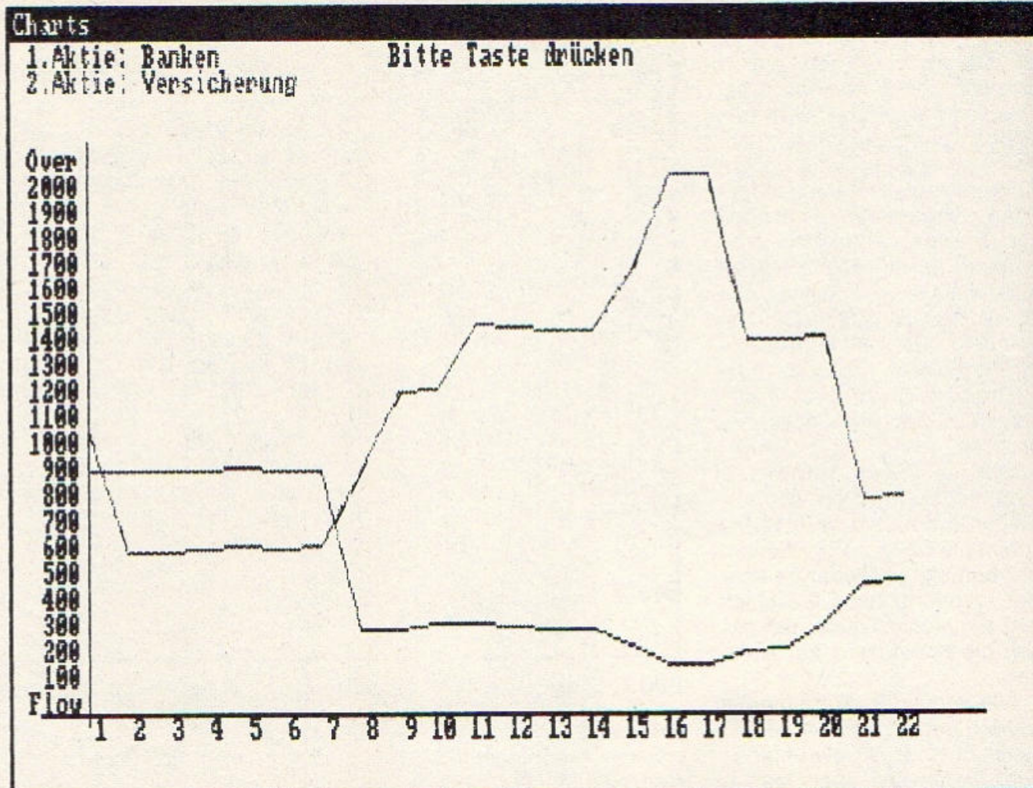


Bild 4. Ausgewählte Aktien können Sie als Grafik ausgeben lassen

Kapitalvergleich

Name	Bargeld	Aktienwert	Brutto	Schulden	Netto
Ivan	861	6364	7225	0	7225
Boesky	191	17303	17494	130	17364

Bitte Taste drücken

Bild 5. Kapitalvergleich gibt Ihnen Informationen über Ihre Mitspieler

So gibt es verschiedene Aktionen, die eine ausgewählte Aktie steigen oder fallen lassen und alle anderen umgekehrt beeinflussen. Wenn Sie Ihren Mitspielern das Leben schwermachen wollen, ist »Manual Activity I« für Sie das Richtige: Hier können Sie selbst eine Aktie bestimmen, die steigen und eine, die sinken soll. Außerdem sind die Veränderungen hier weit größer als bei den zufallsgesteuerten Varianten.

Ebenfalls sehr große Veränderungen erzielen Sie mit »Multiple Action Flood«. Hier wird jede einzelne Aktie entweder stark auf- oder abgewertet. Fast schon harmlos ist dagegen der gelegentlich auftauchende Fiskus, das Finanzamt zeigt sich ungewohnt milde: Steuern werden — im Gegensatz zur Realität — nur auf das Barvermögen erhoben.

Notbremse für Fehler

Jede der zufallsgesteuerten Aktionen kostet Sie einen Handlungspunkt. Sie können Kursveränderungen allerdings auch noch aufhalten, beispielsweise, wenn Ihre Aktien durch eine zufällige Manipulation geschwächt oder die Ihrer Gegner gestärkt würden. In diesem Fall werden Ihnen statt einem gleich drei Handlungspunkte abgezogen. Sie sollten sich also genau überlegen, ob Sie eine Aktion wirklich rückgängig machen wollen. Für die wertvolleren Aktionen, bei denen Sie direkten Einfluß auf die ausgewählten Aktien haben oder die hohe Kursunterschiede mit sich bringen, werden Ihnen immer drei Punkte abgezogen.

Neben diesen Menüpunkten, die aktive Manipulationen zulassen, gibt es die Möglichkeit, Informationen einzuholen. Der Menüpunkt **Börsenkurse** öffnet ein Fenster, in dem die Kurse aller Aktiengesellschaften gezeigt werden. **Ekurs** bedeutet Einstandskurs und dient zur Bewertung des derzeitigen Aktienstands. Wer eine grafische Auswertung der Kursentwicklung sehen möchte, kommt bei **Charts** (Bild 4) auf seine Kosten. Es werden die letzten 22 Kursbewegungen von maximal zwei Aktien gleichzeitig grafisch dargestellt. Hier sollten Sie aber genau wissen, was Sie wollen, denn jede Grafik kostet 60 Mark.

Daneben können Sie sich über die jeweiligen Besitzverhältnisse informieren. **Kapitalliste** zeigt das aktuelle Kapital des Spielers, der am Zug ist, in Barwert und Aktienwert. **Aktienliste** dagegen zeigt Ihren gesamten Aktienbesitz. Um sich Informationen über andere Spieler zu besorgen, muß man Gebühren bezahlen. **Kapitalvergleich** gibt das Kapital aller Spieler aus (Bild 5), **Aktienvergleich** den Besitzstand in Aktien.

Informationen über Ihren eigenen Besitz erhalten Sie kostenlos. Wünschen Sie dagegen Informationen über die anderen Spieler, so müssen Sie 30 Mark (Kapitalvergleich) beziehungsweise 40 Mark (Aktienvergleich) für Gebühren ausgeben.

> **Eingabehinweise**
>
> Tippen Sie Broker mit dem Checksummer ab. Beachten Sie dabei die Anleitung auf Seite 143. Starten Sie jetzt Amiga-Basic. Aktivieren Sie das Ausgabefenster und geben Sie
> `CLEAR ,40000,5000`
> ein. Laden Sie nun mit »Open« Ihr Programm und starten Sie es mit »Start«.

Mit dem Menüpunkt **Sichern** können Sie ein Spiel unterbrechen und den Spielstand auf Diskette sichern. Beim erneuten Start wählen Sie statt **Neues Spiel** einfach **Laden** und spielen dort weiter, wo Sie aufgehört haben.

Sie können hierfür zwei Verfahren anwenden: Wenn Sie für den gespeicherten Spielstand das Directory »Brokerdata« verwenden wollen, brauchen Sie im Programm nichts zu verändern. Sie geben einfach im CLI ein »makedir Brokerdata«. Wollen Sie dagegen ein eigenes Directory verwenden, dann müssen Sie den Pfadnamen bei den gekennzeichneten Stellen im Listing auf die richtige Diskette und Schublade umstellen (Zeile 76, 78, 87 sowie 97, 99 und 108).

Mit dem Menüpunkt **Quit** verlassen Sie Broker, ohne den Spielstand zu sichern.

Tippen Sie das Basic-Programm »Broker« ab und lassen Sie sich faszinieren von der Welt der schnellen Gewinne und Verluste, der Spekulation und der Übernahmen.

Erleben Sie das Auf und Ab der Börse, Hausse und Baisse, Erfolge und Crashs — und das alles, ohne Ihr Bankkonto zu belasten.

Christian Buchner/kn

Programmname:	Broker
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic

```
x00 ' ***************************************************
9S  ' *   BROKER - Das Börsenspiel von Sebastian Peetz   *
up  ' *           (C) 1988 Markt & Technik               *
OR  ' ***************************************************
GP  RANDOMIZE TIMER
i7  OPTION BASE 1
ZD  'Variablen und Screenaufbau
6r  DIM sp$(4),Ak$(13),Ak(4),kap(4),kre(4),ka(4,13),x(13),z(13)
    ,v(13)
tw  DIM st$(13),note$(13),w(4,13),gr(2)
S1  p=0:me2=0:Q=1:req=0:endes=0:no=0
zn  SCREEN 2,640,255,4,2
Sh  WINDOW 2,"BROKER",(0,0)-(630,240),0,2
10  WINDOW OUTPUT 2
XU  ON ERROR GOTO Fehler
fH  FOR i=0 TO 15
5m1  READ a,b,c
3M   PALETTE i,a/15,b/15,c/15
jzO NEXT i
eK  DATA ,,,15,15,15,15,15,,15,,,,15,,,,15,15,,15,8,8,8,,,15
q1  DATA 15,15,15,15,15,,10,5,3,,15,,,,15,15,,15,,15,15
Fd  LOCATE 15,25:PRINT "Bitte warten Sie einen Moment"
Tv  'Aktiendaten lesen
on  RESTORE papiere
OF  FOR i=1 TO 13:READ Ak$(i):NEXT i
6C  'Menüleiste 1
PL  m1:
zt1  RESTORE menua
xw   READ titel:FOR i=1 TO titel
YR   READ unter:FOR j=0 TO unter
uf2   READ statu:IF statu>2 THEN statu=1
z3    IF j=0 AND statu=2 THEN statu=1
Fs    READ a$
J6    MENU i,j,statu,a$
2T1  NEXT j:NEXT i
7DO CLS
SP1  ON MENU GOSUB split
y1   GOTO schl
NUO 'Menüleiste 2
fc  m2:
GB1  RESTORE menub
A9   READ titel:FOR i=1 TO titel
le   READ unter:FOR j=0 TO unter
7s2   READ statu:IF statu>2 THEN statu=1
CG    IF j=0 AND statu=2 THEN statu=1
S5    READ a$
WJ    MENU i,j,statu,a$
8A1  NEXT j:NEXT i:me2=1
KQO CLS
oK  'Mausabfrage zu Menleiste 2
gd1  ON MENU GOSUB split
zd   IF p THEN GOTO Kurse
DO   GOTO schl
tQO 'Hauptprogramm
In  c:
RX1  CLS
Zn   IF hp>0 THEN GOSUB Hpu ELSE IF p THEN GOSUB NoHpu
8GO schl:
Aa1  MENU ON
wM   a$=INKEY$:IF a$="" GOTO schl
HJO split:
XR1  nu=MENU(1):MENU OFF
91   ON MENU(0) GOTO Broker,Boerse,Aktien,Kapakt,Info
oSO Broker:
xp1  ON nu GOTO lad,sav,neue,Ende
YzO Boerse:
bi1  ON nu GOTO Boersenak,Kurse,Charts
X20 Aktien:
Ke1  ON nu GOTO Aktkauf,Aktverk
HIO Kapakt:
V21  ON nu GOTO Kaplist,Kapver,Aktlist,Aktver
eIO Info:
Bw1  ON nu GOTO wech,Uebers,Help
8mO 'Menüpunkt Laden
H41  lad:
g02   WINDOW 4,"Daten laden",(0,10)-(300,180),0,2:COLOR 13,7:CL
      S:LOCATE 2,1
bd    FILES "BrokerData"                    'Directorypfad ge
      gebenenfalls ändern!
6L    PRINT:LINE INPUT "Dateiname: ";datl$
fb    OPEN "Brokerdata/"+datl$ FOR INPUT AS #1  'Dirpfad gegeb
      enenfalls ändern
KA    INPUT #1,s,kapital,spielende,p,hp
8k    FOR i=1 TO s:INPUT #1,sp$(i):INPUT #1,kap(i):INPUT #1,
      kre(i):NEXT i
eF    FOR i=1 TO 13
oL3    INPUT #1,x(i):INPUT #1,v(i):INPUT #1,z(i):INPUT #1,s
       t$(i):INPUT #1,note$(i)
m22   NEXT i
13    FOR j=1 TO s
c93    FOR i=1 TO 13:INPUT #1,w(j,i):INPUT #1,ka(j,i):NEXT i
r82   NEXT j
dQ    CLOSE #1
Ec    FOR i=1 TO 13:we=VAL(MID$(st$(i),1,4))
P73    IF (we<1000 AND we>99)THEN st$(i)=" "+st$(i)
EF     IF we<99 THEN st$(i)="  "+st$(i)
2G2   NEXT i:IF me2 THEN GOTO k
rB    WINDOW CLOSE 4:WINDOW OUTPUT 2
nK    LOCATE 15,25:PRINT "Bitte warten Sie einen Moment":GOTO m
      2
z80 'Menüpunkt Speichern
OC1  sav:
G32   WINDOW 4,"Daten abspeichern",(0,10)-(300,180),0,2:COLOR 1
      3,7:CLS:LOCATE 2,1
2Z    FILES "Brokerdata"                    'Dirpfad gegebene
      nfalls ändern!
qU    PRINT:LINE INPUT"Dateiname: ";dats$
vH    OPEN "BrokerData/"+dats$ FOR OUTPUT AS #1  'siehe oben
4r    PRINT #1,s,kapital,spielende,p,hp
Hr    FOR i=1 TO s:PRINT #1,sp$(i):PRINT #1,kap(i):PRINT #1,
      kre(i):NEXT i
za    FOR i=1 TO 13
Dr3    PRINT #1,x(i):PRINT #1,v(i):PRINT #1,z(i):PRINT #1,s
       t$(i):PRINT #1,note$(i)
7N2   NEXT i
60    FOR j=1 TO s
NF3    FOR i=1 TO 13:PRINT #1,w(j,i):PRINT #1,ka(j,i):NEXT i
CT2   NEXT j
G1    CLOSE #1:KILL "BrokerData/"+dats$+".info" 'siehe oben
VM    WINDOW CLOSE 4:WINDOW OUTPUT 2:GOTO Quit
KAO 'Menüpunkt Neues Spiel
DO1  neue:
Ah2   WINDOW 4,"Neues Spiel",(0,10)-(280,140),0,2:COLOR 15,5:CL
      S
ip1  wie:
en2   LOCATE 2,1:INPUT "Wieviele Mitspieler";s:IF (s<2 OR s>4
      ) THEN wie
ib    PRINT:FOR i=1 TO s
y81  spe:
3Q2   COLOR 1,5
A2    LOCATE (2*i+2),1
eX    PRINT"Name des"i".Spielers:"
yf    COLOR  2,5
E8    LINE INPUT sp$(i):l=LEN(sp$(i)):IF (l<1 OR l>10)THEN sp
      e
Pf1  NEXT i
7A   kapi:
UP2   COLOR 15,5:LOCATE (2*i+3),1:INPUT "Startkapital (500-1200
      0)";kapital
DJ    IF (kapital<500 OR kapital>12000) THEN kapi
FY1  spien:
4h2   LOCATE (2*i+4),1:INPUT "Spielende (>=Kapital*10)";spiele
      nde
Ja    IF(spielende<kapital*10 OR spielende>9999999&)THEN spi
      en
qI    COLOR 2,5:CLS:LOCATE 2,1
P1    FOR i=1 TO s
Us4    PRINT " Spieler";i":  ";sp$(i):PRINT
V52   NEXT i:COLOR 5,2
gH    PRINT:LOCATE ,2:PRINT"Startkapital :";kapital;" DM"
qs    LOCATE ,2:PRINT"Spielende bei:";spielende;"DM"
1J    COLOR 15,5:PRINT:PRINT:INPUT" Alles in Ordung";z$:IF (z$=
      "n" OR z$="N") GOTO Neu
Zt    WINDOW CLOSE 4:WINDOW OUTPUT 2
hY1  anf:
QS2   CLS:LOCATE 15,22:PRINT"Bitte warten - AMIGA generiert Dat
      en"
vL    FOR i=1 TO 13:st$(i)="":NEXT i
bh    FOR ip=1 TO 13:x(ip)=INT(ip/2+.5)*150
```

Listing 1. »Broker« geben Sie mit dem Checksummer (Seite 143) ein

```
LJ3  v(ip)=(ABS((ip<2)+(ip<4)+(ip<6)+(ip<8)+(ip<10)+(ip
     <12))+1)*250
Nr    FOR io=1 TO 23
zN4    h$="0000"+STR$(x(ip))
SD     st$(ip)=st$(ip)+MID$(h$,LEN(h$)-3,4)
b83   NEXT io
gZ    note$(ip)="BBR":z(ip)=INT(ip/2+.5)*150
gE2  NEXT ip
TO   FOR i=1 TO s:kap(i)=kapital:kre(i)=0
7D3   FOR ip=1 TO 13:w(i,ip)=0:ka(i,ip)=0:NEXT ip
r72  NEXT i
1I   p=1:hp=5:IF me2 THEN CLS:GOTO Kurse
UO   GOTO m2
ICO 'Menüpunkt Quit
vJ1 Ende:
h52  MENU RESET
iO   WINDOW CLOSE 2
OU   SCREEN CLOSE 2
ID   END
wkO 'Menüpunkt Börsenaktion
MD  Boersenak:
QJ1  WINDOW 4,"Börsenaktion",(0,10)-(330,235),0,2:CLS
mk   IF hp<3 THEN GOTO Notg
eD   kre(p)=kre(p)+20:r1=INT(RND*13)+1
GP   ON r1 GOTO Divi,HaBa,HaBa,HaBa,HaBa,Flood,HaBa,HaBa,HaBa,H
    aBa,Ma1,Ma2,Fiskus
0b0 'Dividende/Aufwertung
Ky1 Divi:
sJ2  COLOR 0,15:CLS:PRINT:PRINT" Dividende/Aufwertung"
MR    r1=INT(RND*15):IF r1=0 OR r1=14 THEN GOSUB Wahl:GOTO eing
Gw    GOTO wei
mn1 eing:
W82  LOCATE 2,1:INPUT" AG-Nummer Dividende/Aufwertung";r1
7P    IF r1<1 OR r1>13 THEN eing
yP    COLOR 0,15:CLS:PRINT:PRINT" Dividende/Aufwertung"
1s1 wei:
F82   IF x(r1)<z(r1)THEN GOTO wei13
wv    PRINT:PRINT" Dividende 4%":PRINT
xe    PRINT" ----------------------------------------"
kn    PRINT" AG               Diff.   Alt    Neu"
gD    PRINT:PRINT TAB(2)Ak$(r1)TAB(18)INT(x(r1)/25)TAB(26)x(r1)
     TAB(33)x(r1)-INT(x(r1)/25)
0h    PRINT" ----------------------------------------"
cW    PRINT:PRINT" Spieler       Haben    *DM     Gewinn"
VS    FOR ip=1 TO s:LOCATE 13+ip,2
5V3    PRINT sp$(ip)TAB(15)w(ip,r1)TAB(25)INT(x(r1)/25)TAB(32)w
      (ip,r1)*INT(x(r1)/25)
ZK2   NEXT ip:PRINT
7T    LOCATE 19,1:INPUT " Akzeptieren Sie";jn$: IF jn$="n" OR j
     n$="N" THEN hp=hp-3:GOTO wei2
GF    FOR ip=1 TO s:kap(ip)=kap(ip)+w(ip,r1)*INT(x(r1)/25):NEXT
      ip
BC    x(r1)=x(r1)-INT(x(r1)/25):hp=hp-1
rA1  wei2:
mB2   GOTO kk
yI1  wei3:
942   PRINT:PRINT" Aufwertung (Kurs=EKurs)":PRINT
sm    PRINT" ------------------------------------"
Bt    PRINT" AG               Aufw.   Alt    Neu"
C3    PRINT:PRINT TAB(2)Ak$(r1)TAB(18)z(r1)-x(r1)TAB(26)x(r1)TA
     B(33)z(r1)
vp    PRINT" ------------------------------------"
ad    PRINT:PRINT" Spieler       Haben    *DM      Verlust"
kh    FOR ip=1 TO s:LOCATE 13+ip,2
wk3    PRINT sp$(ip)TAB(15)w(ip,r1)TAB(24)z(r1)-x(r1)TAB(32)w(i
      p,r1)*(z(r1)-x(r1))
hF     IF w(ip,r1)=0 THEN wei4
9D     IF kap(ip)>(w(ip,r1)*(z(r1)-x(r1)))THEN kap(ip)=kap(ip)
      -(w(ip,r1)*(z(r1)-x(r1))):GOTO wei4
GP     LOCATE 23,1:PRINT" Achtung, "+sp$(ip)+" !"
5n     PRINT" Verlust Ihrer "+Ak$(r1)+"-Aktien!"
Av     kap(ip)=kap(ip)+w(ip,r1)*x(r1):v(r1)=v(r1)+w(ip,r1):w(ip
      ,r1)=0
Hc1  wei4:
7t2   NEXT ip:x(r1)=z(r1):PRINT:PRINT" Automatische Durchführun
     g!"
Kx    hp=hp-3:GOSUB tast:GOTO kk
tJ0 'Hausse/Baisse
kM  HaBa:
YM1  COLOR 5,2:CLS
YW   IF hp<3 THEN GOTO Notg
fN   IF r1=2 OR r1=7 THEN up=1:dw=1 ELSE IF r1=3 OR r1=8 THEN u
    p=1:dw=2
Jt   IF r1=4 OR r1=9 THEN up=2:dw=1 ELSE up=2:dw=2
C1   r2=INT(RND*13):IF r2=0 OR r2=12 THEN GOSUB Wahl
Nf   IF r1>6 THEN go3
23   IF r2<>0 AND r2<>12 THEN go2
H7  go:
t02  LOCATE 2,1:INPUT " AG-Nummer Hausse";r2:IF r2<1 OR r2>1
    3 THEN go
t11 go2:
Wu2  CLS:PRINT:PRINT" Hausse:":PRINT" -----------------------
    ------------"
vL   PRINT" AG              Diff.   Alt    Neu":PRINT
bH   r1=INT(RND*5)+1
2C   PRINT TAB(2)Ak$(r2)TAB(18)r1*up TAB(26)x(r2)TAB(33)x(r2)+
    r1*up
NH   PRINT" ------------------------------------"
xz   PRINT:PRINT" Baisse:":PRINT" ---------------------------
    --------"
OQ   PRINT" AG              Diff.   Alt    Neu":PRINT
dT   FOR ip=1 TO 13:IF ip=r2 THEN go2a
ZB3   LOCATE 12+ip,2
sT    PRINT Ak$(ip)TAB(18)-r1*dw TAB(26)x(ip)TAB(33)x(ip)-r1*d
     w
Jo1 go2a:
LN2  NEXT ip:PRINT" ------------------------------------"
Np   LOCATE 27,2:INPUT"Akzeptieren Sie";jn$:IF jn$="n" OR jn$=
    "N" THEN hp=hp-3:GOTO kk
nH   FOR ip=1 TO 13
QW3   IF ip=r2 THEN x(ip)=x(ip)+r1*up ELSE x(ip)=x(ip)-r1*dw
Ya2  NEXT ip:hp=hp-1:GOTO kk
EN1 go3:
Zd2  IF r2<>0 AND r2<>12 THEN go5
KU1 go4:
OP2  LOCATE 2,1:INPUT " AG-Nummer Baisse";r2:IF r2<1 OR r2>1
    3 THEN go4
Qb1 go5:
JN2  CLS:PRINT:PRINT" Baisse:":PRINT" -----------------------
    -----------"
Gg   PRINT" AG              Diff.   Alt    Neu":PRINT
wc   r1=INT(RND*5)+1
9T   PRINT TAB(2)Ak$(r2)TAB(18)-r1*dw TAB(26)x(r2)TAB(33)x(r2)
    -r1*dw
ic   PRINT" ------------------------------------"
AU   PRINT:PRINT" Hausse:":PRINT" ---------------------------
    --------"
L1   PRINT" AG              Diff.   Alt    Neu":PRINT
yp   FOR ip=1 TO 13:IF ip=r2 THEN go6
uW3   LOCATE 12+ip,2
3K    PRINT Ak$(ip)TAB(18)r1*up TAB(26)x(ip)TAB(33)x(ip)+r1*up
fr1 go6:
g12  NEXT ip:PRINT" ------------------------------------"
1A   LOCATE 27,2:INPUT"Akzeptieren Sie";jn$:IF jn$="n" OR jn$=
    "N" THEN hp=hp-3:GOTO kk
8c   FOR ip=1 TO 13
mF3   IF ip=r2 THEN x(ip)=x(ip)-r1*dw ELSE x(ip)=x(ip)+r1*up
tv2  NEXT ip:hp=hp-1:GOTO kk
o30 'Multiple Action Flood
wM  Flood:
5p1  COLOR 3,0:CLS
A3   PRINT:PRINT" Multiple Action Flood"
Cn   PRINT:PRINT" ------------------------------------"
yN   PRINT" AG              Diff.   Alt    Neu"
09   FOR ip=1 TO 13:r1=INT(RND*151)-75
Lg2   LOCATE 6+ip,2:PRINT Ak$(ip)TAB(18)r1;"%"TAB(27)x(ip)TAB(3
     4)x(ip)+INT(x(ip)*r1/100)
Qu    x(ip)=x(ip)+INT(x(ip)*r1/100):FOR t=1 TO 3000:NEXT t
4g1  NEXT ip:PRINT" ------------------------------------":PRIN
    T
3Z   PRINT:PRINT" Automatische Durchführung!":hp=hp-3:GOSUB tas
    t:GOTO kk
Zm0 'Manual 1
7a  Ma1:
b81  COLOR 1,14:CLS
n2   PRINT:PRINT" Brokers Manual Activity I":GOSUB Wahl
Ya   so:
Jk   LOCATE 22,1:INPUT" Welche AG-Nummer soll steigen";r1
In   IF r1<1 OR r1>13 THEN so
BV   so2:
pY   LOCATE 23,1:INPUT" Welche AG-Nummer soll fallen ";r2
zn   IF r2<1 OR r2>13 OR r1=r2 THEN so2
JG   COLOR 1,14:CLS
uX   PRINT:PRINT" Brokers Manual Activity I":r3=INT(RND*50)+1
LP   PRINT:PRINT" Es steigt"
NK   PRINT" ------------------------------------"
AW   PRINT" AG              Diff.   Alt    Neu":PRINT
BF   PRINT TAB(2)Ak$(r1)TAB(18)r3;"%"TAB(27)x(r1)TAB(34)INT(x(r
```

```
     1)+x(r1)*r3/100)
283 Sd  x(r1)=INT(x(r1)+x(r1)*r3/100)
284 RO  PRINT" ------------------------------------"
285 F2  PRINT:PRINT" Es fällt:"
286 TQ  PRINT" ------------------------------------"
287 Gc  PRINT" AG              Diff.   Alt   Neu":PRINT
288 na  PRINT TAB(2)Ak$(r2)TAB(18)-r3;"%"TAB(27)x(r2)TAB(34)INT(x(
        r2)-x(r2)*r3/100)
289 s3  x(r2)=INT(x(r2)-x(r2)*r3/100)
290 XU  PRINT" ------------------------------------"
291 A5  PRINT:PRINT:PRINT" Automatische Durchführung!":hp=hp-3:GOS
        UB tast:GOTO kk
292 3H0 'Manual 2
293 a4  Ma2:
294 OX1  COLOR 1,14:CLS
295 cy   PRINT:PRINT" Brokers Manual Activity II"
296 2t   PRINT" -------------------------":PRINT
297 26   PRINT" <1>  Dividende / Aufwertung":PRINT" <2>  Hausse
         /Baisse (1)>  (1/2)<"
298 JW   PRINT" <3>  Hausse/Baisse (1)>  (1/2)<<":PRINT" <4>
           Hausse/Baisse (1)>> (1/2)<"
299 Nu   PRINT" <5>  Hausse/Baisse (1)>> (1/2)<<":PRINT" <6
         >  Multiple Action Flood"
300 bK   PRINT" <7>  Hausse/Baisse (1/2)>  (1)<":PRINT" <8>
         Hausse/Baisse (1/2)>  (1)<<"
301 hh   PRINT" <9>  Hausse/Baisse (1/2)>> (1)<":PRINT" <10>
          Hausse/Baisse (1/2)>> (1)<<"
302 7o   PRINT" <11> Manual Activity I":PRINT" <12> Finanzamt"
303 Gy   LOCATE 22,1:PRINT" Erklärung:  >= normal steigend"
304 Kn   PRINT"              >>= stark  steigend":PRINT"
           <= normal fallend"
305 nn   PRINT"              <<= stark  fallend"
306 Ft   ai:
307 3V2   LOCATE 18,1:INPUT" Börsenaktion-Nummer";r1:IF r1<1 OR r1
          >12 THEN ai
308 nS    ON r1 GOTO Divi,HaBa,HaBa,HaBa,HaBa,Flood,HaBa,HaBa,HaBa,
          HaBa,Ma1,Fiskus
309 Ta0 'Fiskus
310 Kw  Fiskus:
311 q81  max=0:maxi=0
312 01   COLOR 4,0:CLS
313 f1   PRINT:PRINT" Der Fiskus bittet zur Kasse:":PRINT
314 OC   PRINT" Kapitalertragssteuer"
315 uy   PRINT" -------------------":PRINT
316 w5   PRINT" (25% des Barkapitals, sofern dieses "
317 AO   PRINT"  größer ist als der Freibetrag)"
318 rg   PRINT:PRINT" Startkapital:";kapital;"DM"
319 g0   PRINT" Freibetrag:  ";INT(kapital*1.5);"DM"
320 na   PRINT:PRINT:PRINT" Name          Bar DM  Stnetto  Steuer"
         :PRINT
321 sr   FOR ip=1 TO s:LOCATE 15+ip,2
322 WC2   IF kap(ip)>INT(kapital*1.5) THEN max=kap(ip)-INT(kapital
          *1.5):maxi=max/4
323 ZR    PRINT sp$(ip)TAB(14)kap(ip)TAB(23)INT(max)TAB(33)INT(maxi
          )
324 J3    IF kap(ip)>INT(kapital*1.5) THEN kap(ip)=kap(ip)-INT(max
          i):max=0:maxi=0
325 LT1  NEXT ip:PRINT:PRINT
326 oG   PRINT" Automatische Durchführung!":GOSUB tast:GOTO k2
327 i10 'Menüpunkt Börsenkurse
328 Iz  Kurse:
329 cF1  WINDOW 4,"Börsenkurse",(0,10)-(360,175),0,2:COLOR 7,1:CLS
330 XW   PRINT:PRINT " AG                   EKurs    Kurs    Stück"
331 Wn   PRINT" ----------------------------------------"
332 gg   FOR i=1 TO 13:LOCATE 3+i,2:PRINT Ak$(i)TAB(20)z(i)TAB(30)x
         (i)TAB(39)v(i):: NEXT i
333 s4   PRINT:PRINT:PRINT" Bitte Taste drücken":GOTO kkk
334 ak0 'Menüpunkt Charts
335 tK  Charts:
336 4o1  WINDOW 4,"Charts",(0,10)-(630,240),0,2
337 jc   kre(p)=kre(p)+60
338 Y2   ausw:
339 K22   COLOR 0,1:CLS
340 hM    LOCATE 3,2:PRINT"Bitte wählen Sie:"
341 cI    FOR i=1 TO 13:LOCATE 4+i,2
342 w43    PRINT Ak$(i)TAB(18)"<";i;">"
343 H12   NEXT i:LOCATE 24,1
344 ck    PRINT" Zurück zum Menü:   bei 1. < 0 > eingeben"
345 Xd    PRINT" Einzeldarstellung: bei 2. < 0 > eingeben"
346 ab1  eing1:
347 Gg2   LOCATE 20,2:INPUT"1. AG-Nummer (0-13) ";gr(1):IF gr(1)<0
           OR gr(1)>13 THEN eing
348 zd    IF gr(1)=0 THEN kk
349 jg1  eing2:
350 9w2  LOCATE 21,2:INPUT"2. AG-Nummer (0-13) ";gr(2):IF gr(2)<0
          OR gr(2)>13 THEN eing2
351 8J   IF gr(2)=0 THEN ee=1:gr(2)=22 ELSE ee=2
352 T2   COLOR 1,0:CLS:COLOR 4,0:PRINT TAB (2)"1.Aktie: "+Ak$(gr(1
         ))
353 IL   IF gr(2)<>22 THEN COLOR 3,0:PRINT TAB(2)"2.Aktie: "+Ak$
         (gr(2))
354 sN   COLOR 1,0:d=4
355 t0   LOCATE 5,1:PRINT" Over":PRINT" 2000":PRINT" 1900":PRINT"
         1800":PRINT" 1700":PRINT" 1600"
356 2W   PRINT" 1500":PRINT" 1400":PRINT" 1300":PRINT" 1200":PRINT
         " 1100":PRINT" 1000":PRINT"  900"
357 LU   PRINT"  800":PRINT"  700":PRINT"  600":PRINT"  500":PRINT
         "  400":PRINT"  300":PRINT"  200"
358 g4   PRINT"  100":PRINT" Flow":LINE(45,30)-(45,210)
359 dM   LOCATE 27,7:PRINT"1  2  3  4  5  6  7  8  9 10 11 12 13 1
         4 15 16 17 18 19 20 21 22"
360 JU   LINE(15,207)-(600,207)
361 GO   FOR ip=1 TO ee
362 ut3   ilm=0:iln=0:ilv=0:ilb=0:ila=0:ilc=0
363 4G    ilm=VAL(MID$(st$(gr(ip)),5,4)):iln=VAL(MID$(st$(gr(ip)),
          9,4))
364 86    ilv=207-(ilm/12):ilb=207-(iln/12)
365 OL    LINE(45,ilv)-(69,ilb),d
366 6z2  eing3:
367 3Y3   FOR io=3 TO 22
368 YD4    ila=VAL(MID$(st$(gr(ip)),io*4+1,4)):IF ila<20 THEN ein
           g4
369 XM     ilc=207-(ila/12)
370 JX     LINE -(io*24+21,ilc),d
371 Fm3   NEXT io
372 I71  eing4:
373 hT3    d=d-1
374 Lt2   NEXT ip
375 92    LOCATE 1,30:PRINT "Bitte Taste drücken":GOSUB tast2:GOTO
          ausw
376 rV0 'Menüpunkt Aktienkauf
377 LN  Aktkauf:
378 UC1  WINDOW 4,"Aktienkauf",(0,10)-(425,187),0,2:COLOR 13,1:CLS
379 EB   IF hp<2 THEN GOTO Notg
380 YB   PRINT:PRINT" Name: "+sp$(p)+"    Barkapital:";kap(p);"DM"
381 Of   PRINT:PRINT" Welche Aktie möchten Sie kaufen?          <0
         > Keine"
382 kL   PRINT:PRINT" ---------------------------------------------
         -------"
383 Bu   PRINT"   Nr.      AG                    Vorrätig    Kurs"
384 X8   FOR i=1 TO 13
385 Xe2   LOCATE 8+i,2:PRINT TAB(3)"<";i;">"TAB(14)Ak$(i)TAB(34)v
          (i)TAB(47)x(i)
386 fv1  NEXT i
387 86   ein:
388 D12   LOCATE 4,33:INPUT anr:IF anr=0 THEN GOTO ein3 ELSE IF(anr
          <0 OR anr>13)THEN ein
389 I3    COLOR 13,1:CLS
390 gY    PRINT:PRINT" Aktiennummer:";anr:PRINT:PRINT" -----------
          ------------------------------"
391 1z    PRINT" AG               Brief     Kurs     Vorrätig"
392 HL    LOCATE 7,2:PRINT Ak$(anr)TAB(18)z(anr)TAB(28)x(anr)TAB(37
          )v(anr)
393 dK    PRINT" ------------------------------------------"
394 1L    mini=v(anr):maxi=INT(kap(p)/x(anr))
395 Ao    IF mini>maxi THEN maxim=maxi ELSE maxim=mini
396 aU    PRINT:PRINT" Maximale Ankaufsmenge in Stück:";maxim
397 Us    PRINT:INPUT " Ankaufsmenge in Stück";akts
398 JK    IF akts>v(anr) THEN note$(anr)="BG" ELSE IF akts<1 THEN
           note$(anr)="BBR" ELSE GOTO ein2
399 1h    PRINT:PRINT" Ankaufsmenge zu klein!":GOSUB tast:GOTO Aktk
          auf
400 DN1  ein2:
401 Gx2   IF akts*x(anr)>kap(p)THEN PRINT:PRINT" Ihr Kapital reich
          t nicht aus!":GOSUB tast:GOTO Aktkauf
402 r7    an=anr:bn=x(an)
403 tX    PRINT:PRINT:PRINT" Kaufpreis:      ";akts*bn;"DM"
404 Ik    PRINT" Maklergebühr:  ";INT(akts*x(an)/50);"DM"
405 Er    PRINT" -----------------------":PRINT" Effektivsumme: ";
          akts*bn+INT(akts*x(an)/50);"DM"
406 iG    w(p,an)=w(p,an)+akts:note$(an)="GELD":kap(p)=kap(p)-akts*
          bn:v(an)=v(an)-akts
407 79    IF kap(p)<0 THEN kre(p)=kre(p)+ABS(kap(p)):kap(p)=0
408 kT    ka(p,an)=x(an)
409 4q    PRINT:PRINT" Anweisung ausgeführt! Bitte Taste drücken"
410 U1    hp=hp-2:kre(p)=kre(p)+INT((akts*x(an))/50):GOTO kkk
411 Te1  ein3:
412 Nm2   GOTO kk
```

Listing 1. (Fortsetzung)

```
413 dLO 'Menüpunkt Aktienverkauf
414 ML  Aktverk:
415 401  WINDOW 4,"Aktienverkauf",(0,10)-(425,187),0,2:COLOR 3,1:CL
         S
416 pm   IF hp<2 THEN GOTO Notg
417 9m   PRINT:PRINT" Name: "+sp$(p)+"     Barkapital:";kap(p);"DM"
418 38   PRINT:PRINT" Welche Aktie möchten Sie verkaufen?          <
         0> Keine"
419 Lw   PRINT:PRINT" -----------------------------------------------
         -------"
420 oT   PRINT"  Nr.         AG                      Besitz       Kurs"
421 8J   FOR i=1 TO 13
422 sZ2   LOCATE 8+i,2:PRINT TAB(3)"<";i;">"TAB(14)Ak$(i)TAB(34)w
          (p,i)TAB(47)x(i)
423 GW1  NEXT i
424 he   eg1:
425 sS2   LOCATE 4,36:INPUT anr:IF anr=0 THEN GOTO ein3 ELSE IF(anr
          <0 OR anr>13)THEN eg1
426 Vs    IF w(p,anr)=0 OR note$(anr)="GELD" THEN notv ELSE GOTO ja
          ver
427 6h1  notv:
428 su2   PRINT" Verkauf nicht möglich! Bitte Taste drücken":GOSUB
          tast2:GOTO kk
429 LD1  javer:
430 wh2   COLOR 3,1:CLS
431 LD    PRINT:PRINT" Aktiennummer:";anr:PRINT:PRINT" -----------
          --------------------------------"
432 xs    PRINT" AG                  Brief      Kurs      Besitz"
433 Ox    LOCATE 7,2:PRINT Ak$(anr)TAB(18)z(anr)TAB(28)x(anr)TAB(37
          )w(p,anr)
434 Iz    PRINT" ---------------------------------------------"
435 MM    PRINT:PRINT" Kauf der Aktie bei:";ka(p,anr);"für insgesam
          t";ka(p,anr)*w(p,anr);"DM"
436 p4    PRINT:PRINT" Effektivkaufpreis pro Aktie:";ka(p,anr)+INT(
          x(anr)/50);"DM"
437 Pu    PRINT:PRINT:INPUT" Verkaufsmenge in Stück";aktv
438 WZ    IF aktv<1 OR aktv>w(p,anr)THEN PRINT:PRINT" Verkaufsmen
          ge zu klein/groß!":GOSUB tast:GOTO Aktverk
439 kf    IF aktv*x(anr)<1 THEN PRINT:PRINT" Totales Verlustgeschä
          ft !!!":GOSUB tast:GOTO Aktverk
440 xt    an=anr:w(p,an)=w(p,an)-aktv:note$(an)="BRIEF":kap(p)=kap(
          p)+aktv*x(an):v(an)=v(an)+aktv
441 o7    PRINT:PRINT" Verkaufswert: ";x(an)*aktv;"DM":PRINT" Makle
          rgebühr: ";INT((x(an)*aktv)/50);"DM"
442 pm    kre(p)=kre(p)+INT((x(an)*aktv)/50):PRINT:PRINT " Anweisun
          g ausgeführt!  Bitte Taste drücken"
443 6i    hp=hp-2:GOTO kkk
444 3JO 'Menüpunkt Kapitalliste
445 U9  Kaplist:
446 QP1  WINDOW 4,"Kapitalliste",(0,10)-(200,165),0,2:COLOR 12,5:CL
         S
447 AL   GOSUB Aktka
448 Mn   PRINT:PRINT" Name: ";sp$(p)
449 xu   PRINT:PRINT:PRINT" Bargeld:    ";kap(p);"DM"
450 VE   PRINT:PRINT" Aktienwert:";Ak(p);"DM"
451 H3   PRINT:PRINT" ----------------------":PRINT" BRUTTO:      ";
         kap(p)+Ak(p);"DM"
452 1G   PRINT:PRINT" Schulden:  ";kre(p);"DM"
453 6R   PRINT:PRINT" ----------------------":PRINT" NETTO:       ";
         kap(p)+Ak(p)-kre(p);"DM"
454 p1   PRINT:PRINT:PRINT" Bitte Taste drücken":GOTO kkk
455 1vO 'Menüpunkt Kapitalvergleich
456 ZH  Kapver:
457 wJ1  IF no=0 THEN kre(p)=kre(p)+30:g$="Kapitalvergleich"
458 J6   misp:
459 pd2   IF s=2 THEN sp2 ELSE IF s=3 THEN sp3 ELSE sp4
460 EZ1   sp2:
461 7W2   WINDOW 4,g$,(0,10)-(580,90),0,2:COLOR 5,12:CLS
462 mS    GOTO bewe
463 Lh1   sp3:
464 Bu2   WINDOW 4,g$,(0,10)-(580,107),0,2:COLOR 5,12:CLS
465 pV    GOTO bewe
466 Sp1   sp4:
467 Cv2   WINDOW 4,g$,(0,10)-(580,125),0,2:COLOR 5,12:CLS
468 1j    bewe:
469 4N3    PRINT:PRINT" Name                 Bargeld     Aktienwert
           Brutto   Schulden     Netto"
470 ic     PRINT" ------------------------------------------------
           ---------------------"
471 YJ     GOSUB Aktka
472 Od4     FOR ip=1 TO s
473 7t5      PRINT TAB(2)sp$(ip)TAB(20)kap(ip)TAB(31)Ak(ip)TAB(44)k
             ap(ip)+Ak(ip)TAB(53)kre(ip)TAB(65)kap(ip)+Ak(ip)-kre(ip
             )
474 h64      PRINT:NEXT ip
475 cu3     PRINT:PRINT" Bitte Taste drücken":no=0
476 Fr      GOTO kkk
477 n30 'Menüpunkt Aktienliste
478 RA  Aktlist:
479 eX1  WINDOW 4,"Aktienliste",(0,10)-(420,235),0,2:COLOR 2,0:CLS
480 hs   GOSUB Aktka
481 tK   PRINT:PRINT" Name: ";sp$(p)
482 PM   PRINT:PRINT" Unverbrieftes Kapital: ";kap(p);"DM"
483 TN   PRINT" Verbrieftes Kapital  : ";Ak(p);"DM"
484 Iz   PRINT" -------------------------------------------------"
485 pc   PRINT" AG                    Umsz.      Habe     Kurs":PRINT
486 Qm   FOR i=1 TO 13:LOCATE 8+i,2:PRINT Ak$(i)TAB(22)note$(i)TAB(
         31)w(p,i)TAB(40)x(i)
487 ps   NEXT i:PRINT" -------------------------------------------
         ----"
488 rO   PRINT:PRINT" Gesamtkapital: ";kap(p)+Ak(p);"DM"
489 Oa   PRINT:PRINT:PRINT" Bitte Taste drücken":GOTO kkk
490 IkO 'Menüpunkt Aktienvergleich
491 YK  Aktver:
492 231 kre(p)=kre(p)+40
493 IR   IF s=2 THEN s2 ELSE IF s=3 THEN s3 ELSE s4
494 CF   s2:
495 Gw2   WINDOW 4,"Aktienvergleich",(0,10)-(360,170),0,2:COLOR 0,2
          :CLS
496 6o    PRINT:PRINT" AG"TAB(21)sp$(1)TAB(35)sp$(2)
497 CT    PRINT" ------------------------------------------"
498 w6    FOR ip=1 TO 13:LOCATE 3+ip,2
499 1s3    PRINT Ak$(ip)TAB(20)w(1,ip)TAB(34)w(2,ip)
500 QV2   NEXT ip:PRINT:PRINT:PRINT" Bitte Taste drücken":GOTO kkk
501 MQ1  s3:
502 SA2   WINDOW 4,"Aktienvergleich",(0,10)-(470,170),0,2:COLOR 0,2
          :CLS
503 kT    PRINT:PRINT" AG"TAB(21)sp$(1)TAB(35)sp$(2)TAB(48)sp$(3)
504 ga    PRINT" ---------------------------------------------
          -----"
505 3D    FOR ip=1 TO 13:LOCATE 3+ip,2
506 e13    PRINT Ak$(ip)TAB(20)w(1,ip)TAB(34)w(2,ip)TAB(47)w(3,ip)
507 Xc2   NEXT ip:PRINT:PRINT:PRINT" Bitte Taste drücken":GOTO kkk
508 Wb1  s4:
509 bK2   WINDOW 4,"Aktienvergleich",(0,10)-(570,170),0,2:COLOR 0,2
          :CLS
510 5p    PRINT:PRINT" AG"TAB(21)sp$(1)TAB(35)sp$(2)TAB(48)sp$(3)TA
          B(61)sp$(4)
511 qK    PRINT" ---------------------------------------------
          ------------------"
512 AK    FOR ip=1 TO 13:LOCATE 3+ip,2
513 xP3    PRINT Ak$(ip)TAB(20)w(1,ip)TAB(34)w(2,ip)TAB(47)w(3,ip)T
           AB(60)w(4,ip)
514 ej2   NEXT ip:PRINT:PRINT:PRINT" Bitte Taste drücken":GOTO kkk
515 3RO 'Menüpunkt Nächster
516 7i  Nexte:
517 rR1  CALL requester(" Spielerwechsel?","Ja","Nein")
518 yx   LOCATE 18,2
519 xy   IF req=2 THEN WINDOW OUTPUT 2:req=0:GOTO c
520 Z8   IF req=1 THEN wech
521 2C   GOTO Nexte
522 DI   wech:
523 fe2   WINDOW 4,"Spielerwechsel",(0,10)-(330,240),0,2:COLOR 1,5:
          CLS
524 L4    PRINT:PRINT" Bitte warten Sie: AMIGA rechnet!":hp=5
525 kP    IF kre(p)>0 AND kap(p)>kre(p) THEN GOSUB Schulden
526 Au    IF kre(p)>0 AND kap(p)<kre(p) THEN GOSUB Zinsen
527 Tr    GOSUB Aktka:krelim=kapital/2
528 rS    FOR i=1 TO 13
529 AM3    IF x(i)<50 THEN GOSUB Low ELSE IF x(i)>2000 THEN GOSUB
           High
530 mu     h$=STR$(x(i)):IF LEN(h$)=4 THEN h$=" "+h$ ELSE IF LEN(h$
           )=3 THEN h$="  "+h$
531 nt     st$(i)=MID$(st$(i),5,88)+MID$(h$,2,4)
532 PU     IF note$(i)="GELD" THEN note$(i)="BZ"
533 2I2   NEXT i
534 OX    IF (kap(p)+Ak(p))>=spielende AND kre(p)=0 THEN endes=1
535 fH    IF ((kap(p)+Ak(p)-kre(p)<100)OR kre(p)>krelim)THEN ende
          s=1
536 cO    p=p+1:IF p>s THEN p=1
537 1y    IF endes THEN Spielend
538 h1    PRINT:PRINT:PRINT" Nächster Spieler ist: ";sp$(p)
539 GO    PRINT:PRINT" Bitte Taste drücken":GOTO kkk
540 LJ1  Spielend:
541 JX2   GOSUB tast:WINDOW CLOSE 4:WINDOW OUTPUT 2
542 gH    g$="Schlußberechnung":CLS:GOTO misp
543 BNO 'Menüpunkt Uebersicht
544 W8  Uebers:
```

```
545 gw1  WINDOW 4,"Uebersicht",(0,10)-(250,140),0,2:COLOR 5,6:CLS
546 WO   PRINT:FOR i=1 TO s:PRINT" Spieler";i;":    ";sp$(i):PRINT:N
         EXT i
547 h4   PRINT:COLOR 6,5:LOCATE ,2:PRINT "Startkapital :";kapital;"
          DM"
548 wI   LOCATE ,2:PRINT"Spielende bei:";spielende;"DM":COLOR 5,6:P
         RINT
549 IW   GOSUB tast:GOTO kk
550 5tO 'Menüpunkt Help
551 Oz  Help:
552 fD1  WINDOW 4,"BROKER - Das Börsenspiel von Sebastian Peetz",(0
         ,10)-(630,200),0,2
553 TO   COLOR 5,15:CLS
554 Iv   PRINT
555 8S   PRINT " Menüpunkt                            Erklärung
                             DM] HP"
556 zT   PRINT "-----------------------------------------------------
         ---------------------------"
557 Fx   PRINT "Laden                          Ein Spiel wird von Disk gel
         aden                       ]"
558 Bv   PRINT "Speichern                      Ein Spiel wird abgespeicher
         t                          ]"
559 zP   PRINT "Neues Spiel                    Es wird ein neues Spiel beg
         onnen                      ]"
560 xH   PRINT "Quit                           Spielende ohne die Spielstä
         nde abzuspeichern          ]"
561 41   PRINT "Börsenaktion                   Der Spieler kann aktiv die
         Börsenkurse             20] 1-3"
562 uS   PRINT "                               durch geschicktes Handeln b
         eeinflussen                ]"
563 Mr   PRINT "Börsenkurse                    Es werden die aktuellen Bör
         senkurse angezeigt      - ] -"
564 VJ   PRINT "Charts                         Aktienkurse können graphisc
         h dargestellt werden    60] -"
565 MJ   PRINT "Aktienkauf                     Aktien können zum Tageskurs
          gekauft werden         5%] 2"
566 H1   PRINT "Aktienverk.                    Aktien können zum Tageskurs
          verkauft werden        5%] 2"
567 3G   PRINT "Kapitalliste                   Eigenes Kapital (Aktienkapi
         tal und bar)            - ] -"
568 iw   PRINT "Kapitalvergl.                  Kapital aller Spieler
                                 30] -"
569 m6   PRINT "Aktienliste                    Auflistung der eigenen Akti
         en (Kurse,Stück)        - ] -"
570 WZ   PRINT "Aktienvergl.                   Aktien aller Spieler (Stück
         )                       40] -"
571 aJ   PRINT "Nächster                       Der nächste Spieler ist an
         der Reihe                  ]"
572 XQ   PRINT "Uebersicht                     Alle wichtigen Daten zum Sp
         iel im überblick           ]"
573 IT   PRINT "Info                           Diese Seite
                                    ]"
574 rb   PRINT:PRINT "Bitte Taste drücken":GOTO kkk
575 FPO 'Unterprogramme
576 jf  'Subprogramme für Menüpunkt Börsenaktion/Hausse-Baisse
577 WA  Wahl:
578 hW1  LOCATE 4,1:PRINT" -----------------------------------------
         "
579 bn   PRINT"   Nr.    AG                 Habe    Kurs"
580 hI   FOR i=1 TO 13
581 zl2   LOCATE 6+i,2:PRINT "<";i;">"TAB(11)Ak$(i)TAB(28)w(p,i)T
          AB(36)x(i)
582 391  NEXT i:PRINT" ---------------------------------------"
583 1N   RETURN
584 h20 'Subprogramme für Menüpunkt Nächster
585 qG  Schulden:
586 WR1  PRINT:PRINT:PRINT" Schulden:":PRINT" ---------"
587 99   PRINT:PRINT " "+sp$(p)+" muß ";kre(p);"DM  bezahlen."
588 EN   kap(p)=kap(p)-kre(p):kre(p)=0
589 rT   RETURN
590 620 Zinsen:
591 t01  PRINT:PRINT:PRINT"  Schulden: ";kre(p);"DM":PRINT"+ Zinsen
         :    ";CINT(kre(p)/10);"DM"
592 K1   PRINT"  -----------------":PRINT:PRINT"  Gesamt:    ";kre(p
         )+CINT(kre(p)/10);"DM"
593 Io   kre(p)=kre(p)+CINT(kre(p)/10)
594 wY   RETURN
595 YMO Low:
596 QJ1  PRINT:PRINT:PRINT" Aufwertung:":PRINT" -----------":PRINT
597 gs   PRINT" Kurswert zu niedrig bei:  "+Ak$(i)+" !"
598 mY   PRINT" Aufwertung je Aktie     :";(z(i)-x(i));"DM":PRINT
599 Rg   FOR ip=1 TO s
600 9I2   IF kap(ip)<w(ip,i)*(z(i)-x(i)) THEN verl
601 9n    PRINT TAB(2)sp$(ip)TAB(13)"muß"TAB(17)w(ip,i)*(z(i)-x(i))
          ;"DM zahlen."
602 kD    kap(ip)=kap(ip)-w(ip,i)*(z(i)-x(i))
603 MG    GOTO weiter
604 nA1  verl:
605 sk3    PRINT TAB(2)sp$(ip)TAB(13)"- AKTIENVERLUST -"
606 FX     kap(ip)=kap(ip)+w(ip,i)*x(i):v(i)=v(i)+w(ip,i):w(ip,i)=0
607 OL1  weiter:
608 7f2   NEXT ip
609 1v    x(i)=z(i)
610 07    PRINT:PRINT" - TASTE -":GOSUB tast2
611 Dp1  RETURN
612 b30 High:
613 Jo1  PRINT:PRINT:PRINT" Dividende:":PRINT" ----------":PRINT
614 pP   PRINT" Kurswert zu hoch bei:  "+Ak$(i)+" !"
615 7S   PRINT" Auszahlung je Aktie : ";(x(i)-2000);"DM":PRINT
616 ix   FOR ip=1 TO s
617 OK2   PRINT TAB(2)sp$(ip)TAB(13)"erhält"TAB(20);(w(ip,i)*(x(i)-
          2000));"DM"
618 wy    kap(ip)=kap(ip)+(w(ip,i)*(x(i)-2000))
619 Iq1  NEXT ip
620 9D   x(i)=2000
621 BI   PRINT:PRINT" - TASTE -":GOSUB tast2
622 00   RETURN
623 k50 'Sonstige Subprogramme
624 Mt  Aktka:
625 r61  FOR ip=1 TO s
626 fQ2   Ak(ip)=0
627 S33    FOR i=1 TO 13
628 h14     Ak(ip)=Ak(ip)+x(i)*w(ip,i)
629 aq3    NEXT i
630 wZ1  NEXT ip:RETURN
631 9zO Notg:
632 hM1  PRINT:PRINT:PRINT" Sie haben nicht genug Handlungspunkte!"
633 mW   PRINT:PRINT" Bitte Taste drücken":GOTO kkk
634 yhO Hpu:
635 7f1  LOCATE 11,24:PRINT "Spieler: ";sp$(p)
636 3Y   LINE(175,90)-(460,125),3,BF
637 5U   LOCATE 14,23:COLOR 8,3
638 6V   PRINT" Sie haben noch";hp;"Handlungspunkte! "
639 4M   COLOR 1,0
640 gI   RETURN
641 kdO NoHpu:
642 Em1  LOCATE 11,24:PRINT "Spieler: ";sp$(p)
643 Go   LINE(175,90)-(490,125),3,BF
644 Cb   LOCATE 14,23:COLOR 8,3
645 EL   PRINT" Sie haben keine Handlungspunkte mehr! "
646 BT   COLOR 1,0
647 nP   RETURN
648 NmO Fehler:
649 by1  WINDOW 5,"ERROR - ERROR - ERROR",(310,10)-(610,80),0,2:COL
         OR 1,5:CLS
650 13   PRINT:PRINT" Es ist folgender Fehler aufgetreten:":PRINT
651 Kc   IF ERR=53 THEN PRINT" DATEI NICHT GEFUNDEN !":GOTO dudi
652 4Y   IF ERR=61 THEN PRINT" DISKETTE IST VOLL !":GOTO dudi
653 my   IF ERR=64 THEN PRINT" FALSCHER DATEINAME !":GOTO dudi
654 iJ   IF ERR=67 THEN PRINT" ZUVIELE DATEIEN !":GOTO dudi
655 Vh   IF ERR=70 THEN PRINT" DISKETTE SCHREIBGESCHÜTZT !":GOTO du
         di
656 Wi   IF ERR=74 THEN PRINT" UNBEKANNTE DISKETTE":GOTO dudi
657 Xx   PRINT " UNBEKANNTER FEHLERTYP!"
658 im    dudi:
659 QU2   PRINT:GOSUB tast:WINDOW CLOSE 5:RESUME kk
660 SJO 'Häufig verwendete Sub-Programme
661 Ly  k:
662 2k1  WINDOW CLOSE 4
663 CU   WINDOW OUTPUT 2
664 Yo   GOTO Kurse
665 hcO k2:
666 6o1  WINDOW CLOSE 4
667 GY   WINDOW OUTPUT 2
668 VN   no=1:GOTO Kapver
669 WMO kk:
670 As1  WINDOW CLOSE 4
671 Kc   WINDOW OUTPUT 2
672 gk   GOTO c
673 qtO kkk:
674 JW1  IF INKEY$="" GOTO kkk
675 Fx   WINDOW CLOSE 4
676 Ph   WINDOW OUTPUT 2
677 Oc   IF endes THEN Ende
678 mq   GOTO c
679 SsO tast:
```

Listing 1. (Fortsetzung)

```
680 2M1  PRINT:PRINT " Bitte Taste drücken"
681 jJ   WHILE INKEY$=""
682 RF   WEND
683 Nz   RETURN
684 bx0 tast2:
685 nN1  WHILE INKEY$=""
686 VJ   WEND
687 R3   RETURN
688 tv0 'Diverse Daten
689 ER  papiere:
690 Z91  DATA Bauwerte,Textil,Brauerei,Kosmetik,Elektrizität,Chemie
         ,Pharmazie
691 Ic   DATA Maschinenbau,Kaufhaus,Computer,Automobil,Banken,Versi
         cherung
692 ODO menua:
693 dS1  DATA 5
694 cQ   DATA 4
695 cz   DATA 1,Broker
696 rh   DATA 1,"Laden      "
697 HR   DATA 0,"Speichern  "
698 wQ   DATA 1,"Neues Spiel"
699 6n   DATA 1,"Quit       "
700 gT   DATA 3
701 GI   DATA 0,Börse
702 G5   DATA 0,"Börsenaktion"
703 16   DATA 0,"Börsenkurse "
704 ON   DATA 0,"Charts      "
705 jV   DATA 2
706 I6   DATA 0,Aktien
707 Oh   DATA 0,"Aktienkauf   "
708 Vt   DATA 0,"Aktienverkauf"
709 rf   DATA 4
710 64   DATA 0,Kapital/Aktienwerte
711 FM   DATA 0,"Kapitalliste       "
712 qu   DATA 0,"Kapitalvergleich   "
713 Dt   DATA 0,"Aktienliste        "
714 fU   DATA 0,"Aktienvergleich    "
715 vi   DATA 3
716 El   DATA 1,Information
717 7H   DATA 0,"Nächster   "
718 HP   DATA 0,"Uebersicht"
719 1k   DATA 1,"Help       "
720 Yh0 menub:
721 5u1  DATA 5
722 4s   DATA 4
723 4R   DATA 1,Broker
724 J9   DATA 1,"Laden      "
725 1w   DATA 1,"Speichern  "
726 Os   DATA 1,"Neues Spiel"
727 YF   DATA 1,"Quit       "
728 8v   DATA 3
729 kn   DATA 1,Börse
730 ka   DATA 1,"Börsenaktion"
731 Fb   DATA 1,"Börsenkurse "
732 ss   DATA 1,"Charts      "
733 Bx   DATA 2
734 mb   DATA 1,Aktien
735 sC   DATA 1,"Aktienkauf   "
736 z0   DATA 1,"Aktienverkauf"
737 J7   DATA 4
738 aZ   DATA 1,Kapital/Aktienwerte
739 jr   DATA 1,"Kapitalliste       "
740 KP   DATA 1,"Kapitalvergleich   "
741 h0   DATA 1,"Aktienliste        "
742 9z   DATA 1,"Aktienvergleich    "
743 NA   DATA 3
744 gD   DATA 1,Information
745 bm   DATA 1,"Nächster   "
746 lu   DATA 1,"Uebersicht"
747 TC   DATA 1,"Help       "
(C) 1988 M&T
```

Listing 1. (Schluß)

Packen wir's

Wollen Sie mehr Programme auf einer Diskette unterbringen? Die Lösung heißt »MasterCruncher«. Er packt alles.

AmigaBasic mit 103484 oder mit 71412 Byte? Die Entscheidung fällt nicht schwer: die kürzere Version ist besser. Ein kompletter Aztec-C-Compiler inklusive Source-Debugger mit 263388 oder mit 156292 Byte? Auch hier ist das Ergebnis klar. Wie diese Beispiele deutlich machen, können Sie erstaunlich viel Speicherplatz auf Ihren Disketten einsparen, wenn Sie unser Programm des Monats besitzen. In der Regel handelt es sich um 30 bis 50 Prozent der Originallänge. Die Funktionsweise von »MasterCruncher« ist im Prinzip einfach. Programme und Dateien enthalten gleiche Bytefolgen. Wenn diese Sequenzen zu kürzeren zusammengefaßt werden, heißt dies »packen« oder »crunchen«. Dabei muß man zwischen drei Typen von Dateien unterscheiden:

- verschiebbare Programme,
- Daten,
- nicht verschiebbare Programme.

Im Normalfall gehören die Programme auf dem Amiga zur ersten Kategorie. Nur wenige Ausnahmen fallen in die dritte Gruppe. Der Unterschied liegt darin, daß bei den nicht verschiebbaren Programmen ein bestimmter Speicherbereich verwendet wird. Gepackte Programme lassen sich genau wie vorher von der Workbench oder dem CLI aufrufen. Zu diesem Zweck muß die gepackte Datei im Speicher zuerst entpackt werden. Das erledigt MasterCruncher automatisch. Vor die komprimierten Daten wird ein Programm geschrieben, das diese Aufgabe übernimmt. Bei langen Programmen verringert sich sogar die Ladezeit von Diskette, trotz des notwendigen Entpackens.

Bei Daten ist eine andere Vorgehensweise nötig. Da hier keine Programmverzweigung auf die Daten erfolgt, können sie nicht im Speicher entpackt werden. Komprimierte Daten werden mit MasterCruncher geladen und anschließend ungepackt auf Diskette gespeichert. Dann können Sie wiederverwendet werden.

MasterCruncher liegt in Form eines DATA-Laders vor, damit ihn jeder Amiga-Besitzer abtippen kann. Das Amiga-Basic-Programm (siehe Listing) generiert auf der Diskette das lauffähige Programm. Vor dem Laden des Listings müssen Sie allerdings den Befehl »CLEAR ,40000« eingeben. Der Aufruf

```
MasterCruncher
```

startet das Programm vom CLI aus. Vor der Benutzung von der Workbench aus muß noch ein passendes Programmsymbol (Icon) kopiert werden oder mit IconEd ein eigenes gezeichnet werden. Der Typ des Icons muß Tool sein. Dann kann man MasterCruncher durch Doppelklick auf das Programmsymbol starten.

Da MasterCruncher selbst gepackt wurde (statt 14036 nur noch 9328 Byte), flackert kurz der Bildschirm und dann erscheint das Bedienungsfeld (siehe Bild). Betrachten wir nun die normale Vorgehensweise.

LOAD FILE/SAVE FILE

Zunächst laden Sie die zu packende Datei. Dazu klicken Sie einmal auf den Schalter (Gadget) »LOAD FILE«. Nach Betätigung erscheint ein Datei-Requester, mit dessen Hilfe Sie die Datei auswählen. Oberhalb der Einträge finden Sie ein Textfeld (Stringgadget) für den Pfad. Nach Anklicken erscheint ein Cursor zur Eingabe des neuen Pfads. Die Pfade »DF0:« und »DF1:« erreichen Sie durch Anklicken der entsprechenden Schalter. Unterhalb der Einträge sehen Sie ein Stringgadget, in dem der Name der Datei eingegeben

```
----==>> MASTERCRUNCHER V3.00 <<==----
CODED BY ROBERT FRAHM & MICHAEL CREMER
   COPYRIGHT BY MARKT & TECHNIK
CRUNCHING MODE....: 0000111F    | OK
DECRUNCH CONTROL..: 00DFF180    | HEX: ON
RECRUNCH TO.......: --------    | RCA: ON
JUMP TO...........: --------    | RESET
SAVE CONTROL......: RELOCABLE
SOURCEFILE      START ADDRESS..: $00C28168
SOURCEFILE      END   ADDRESS..: $00C28AD8
SOURCEFILE      LENGTH.........: $00000970
DESTINATION     START ADDRESS..: $00C27D68
DESTINATION     END   ADDRESS..: $00C27D68
DESTINATION     LENGTH.........: $000003A8
MEMORY DUMP AT SOURCE PTR. : $00C28168
00:00 00 03 F3 00 00 00 00 00 00 00 09
0C:00 00 00 00 00 00 00 08 00 00 00 83
18:00 00 00 6B 00 00 00 65 00 00 00 1D
24:00 00 00 06 00 00 00 00 00 00 00 20
30:00 00 00 1F 00 00 00 00 00 00 03 E9
```

Das übersichtliche Bedienungsfeld von »MasterCruncher«

werden kann. Durch die zwei Pfeile können Sie in der Liste der Einträge auf- und abscrollen. Mit dem »CANCEL«-Schalter brechen Sie den Ladevorgang ab. Durch Anklicken von »OK« starten Sie das Laden.

Sollte ein Fehler auftreten, wird dies im Hauptmenü angezeigt. Ein möglicher Fehler ist, daß das Programm mit der Overlay-Technik arbeitet. Solche Programme (Deluxe Paint II, Butcher) können nicht gepackt werden. Zum Glück gibt es aber nicht viele davon.

Wenn Sie eine gepackte Datei laden, flackert kurz der Bildschirm auf und die Datei wird entpackt. Dann benutzen Sie »SAVE FILE«, um die Datei ungepackt zu speichern. Sollte die Datei gegen Entpacken geschützt sein (dazu später mehr), erscheint die Meldung:

```
ERROR... UNABLE to REPACK FILE...
```

»SAVE FILE« dient zum Speichern von gepackten oder ungepackten Daten. Die Bedienung ist dieselbe wie beim Laden.

PACK FILE

Die vorher geladene Datei wird komprimiert. Befinden sich keine Daten im Speicher, unterbricht MasterCruncher den Vorgang. Verläuft alles normal, erscheint die Meldung:

```
PLEASE WAIT... PACKING... XXXXXX
```

Die Zahl (XXXXXX) am Ende gibt die Länge des noch zu packenden Bereichs an. Der Text verschwindet, sobald der Vorgang beendet ist. Ein Klick mit der linken Maustaste unterbricht jederzeit das Packen. Während der Arbeit ist das Multitasking des Amiga abgeschaltet. Es kann also kein anderes Programm laufen.

PREFERENCES

In diesem Menü können einige Voreinstellungen verändert werden, die den Pack- und Entpackvorgang betreffen. Der »CRUNCHING MODE« bestimmt die Länge des zu untersuchenden Speicherbereichs. Hier sind Werte von $20 bis $111F erlaubt. Je größer der Wert, um so länger dauert das Packen und um so kürzer wird die Datei. Die Voreinstellung ($111F) ist meist sinnvoll.

Bei »DECRUNCH CONTROL« steht eine Adresse, die während des Entpackens verändert wird. Mit dem eingetragenen Wert $DFF180 beeinflußt MasterCruncher die Hintergrundfarbe. Für die Interessierten hier die anderen möglichen Adressen:

```
$DFF182: Farbregister 1
$DFF183: Farbregister 2
   .
   .
$DFF1BE: Farbregister 31
```

Bei Verwendung der höheren Farbregister ist es möglich, daß auf dem Bildschirm nichts zu sehen ist. Es empfiehlt sich, den vorgegebenen Wert zu übernehmen.

»RECRUNCH TO« ist nur für nicht verschiebbare Programme interessant. Hier wird die absolute Adresse angegeben, an die das Programm entpackt wird. Für diese Programme ist auch »JUMP IN« wichtig. Es bestimmt die Adresse, an die MasterCruncher nach dem Entpacken verzweigt. Nach der Auswahl dieser zwei Schalter, schaltet sich »SAVE CONTROL« automatisch auf »ADRESS RECRUNCHER« um. Die zwei anderen Einstellungen sind »RELOCABLE« (verschiebbare Programme) und »DATA + INFO HEADER« (Daten). Reine Daten sind natürlich nicht vom CLI aus aufrufbar. Die verschiedenen Möglichkeiten erreichen Sie durch mehrmaliges Anklicken des Schalters SAVE CONTROL.

Der Hexdump kann wahlweise mit »HEX« ein- und ausgeschaltet werden. Ohne diese Ausgabe erfolgt das Packen schneller.

»RCA« steht für »repackable« (entpackbar). Steht dieser Schalter auf OFF, kann die gepackte Datei nicht mehr entpackt werden.

»RESET« stellt den Ursprungszustand wieder her.

Mit »OK« gelangen Sie ins Hauptmenü.

Nach getaner Arbeit, bei der bei langen Dateien eine Tasse Kaffee angebracht ist, verlassen Sie MasterCruncher mit dem Schalter »EXIT TO ...«. Am Ende steht entweder »CLI« oder »WORKBENCH«, je nachdem, wie das Programm aufgerufen wurde.

Durch das Programm »MasterCruncher« verringert sich in der Regel die Ladezeit und Sie sparen wichtigen Speicherplatz auf Ihren Disketten.

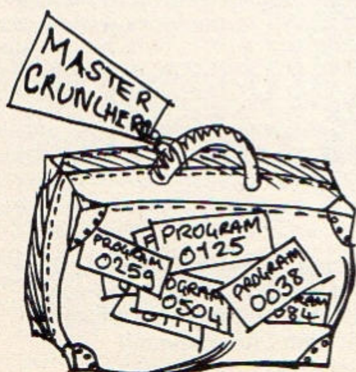
R. Frahm/M. Cremer/R. Beaupoil/kn

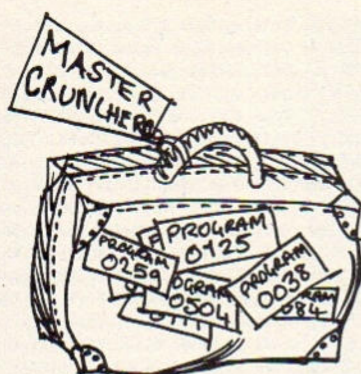
Programmname:	MasterCruncher_Gen
Computer:	A500, A1000, A2000 mit Kickstart 1.2 & 1.3
Sprache:	Amiga-Basic 1.2
Bemerkung:	Erzeugt fertiges Programm

```
Programmautor: Michael Cremer / Robert Frahm
--------------------------------------------
Om0 REM Generiert lauffähiges Programm
ag  CLS
1G  OPEN "MasterCruncher" FOR OUTPUT A
    S 1
BS  READ anz
oa  FOR i=1 TO anz
3n1  READ h$
yB2   wert1=ASC(LEFT$(h$,1))
bP    IF wert1>64 THEN wert1=wert1-87
       ELSE wert1=wert1-48
FI    wert1=wert1*16
7c    wert2=ASC(RIGHT$(h$,1))
wp    IF wert2>64 THEN wert2=wert2-87
       ELSE wert2=wert2-48
Pi    wert=wert1+wert2
9G    PRINT #1,CHR$(wert);
JOO NEXT
3n  CLOSE 1
qE  KILL "MasterCruncher.info"
1w  END
zd  Werte:
qi  DATA 9364
qi  DATA 00,00,03,f3,00,00,00,00,00,00
nt  DATA 00,03,00,00,00,00,00,00,00,02
5m  DATA 00,00,00,d8,00,00,08,38,00,00
PG  DATA 0d,b1,00,00,03,e9,00,00,00,d8
60  DATA 48,e7,ff,fe,4b,fa,03,38,41,fa
xW  DATA ff,f2,22,50,d3,c9,d3,c9,24,51
Sk  DATA 58,89,2b,49,00,14,d5,ca,d5,ca
vV  DATA 58,8a,2b,4a,00,18,48,e7,ff,fe
dQ  DATA 61,00,01,bc,4c,df,7f,ff,22,6d
ne  DATA 00,14,51,89,20,11,2c,78,00,04
B1  DATA 4e,ae,ff,2e,26,6d,00,18,50,8b
VU  DATA 20,1b,2e,00,e7,88,22,3c,00,01
CL  DATA 00,00,4e,ae,ff,3a,2b,40,00,0c
Rk  DATA 50,8b,20,07,53,80,22,6d,00,0c
kN  DATA 22,1b,58,89,54,81,22,c1,51,c8
4F  DATA ff,f6,2f,0b,2c,07,53,86,26,6d
vC  DATA 00,0c,28,4b,4a,9b,22,3c,00,01
Ah  DATA 00,00,20,1b,58,8b,08,00,00,1e
GN  DATA 67,04,08,c1,00,01,e5,88,4e,ae
zq  DATA ff,3a,28,c0,58,8c,51,ce,ff,e0
Mq  DATA 26,5f,7c,00,7a,00,0c,6b,03,e9
B1  DATA 00,02,67,00,00,9a,0c,6b,03,ea
ui  DATA 00,02,67,00,00,90,0c,6b,03,eb
gH  DATA 00,02,67,00,00,a8,0c,6b,03,ec
SA  DATA 00,02,67,00,00,b0,0c,6b,03,f2
1K  DATA 00,02,4e,71,4a,9b,7a,00,52,86
sN  DATA be,86,6e,c6,2f,07,43,fa,ff,18
```

Listing. »MasterCruncher_Gen« erzeugt den lauffähigen Packer »MasterCruncher« auf Diskette. Mit dem Checksummer (Seite 143) eingeben.

```
47 8d   DATA 20,6d,00,0c,20,10,58,80,e4,88
48 42   DATA 22,80,52,87,53,87,67,1a,22,50
49 6E   DATA 20,28,00,04,e5,88,22,80,20,28
50 T3   DATA 00,08,58,80,e4,88,23,40,00,04
51 j6   DATA 50,88,60,e2,2e,1f,42,a9,00,04
52 d7   DATA 43,fa,00,2a,20,6d,00,0c,20,50
53 kB   DATA 50,88,22,88,22,6d,00,0c,20,07
54 ae   DATA e7,88,4e,ae,ff,2e,22,6d,00,18
55 MM   DATA 51,89,20,11,4e,ae,ff,2e,4c,df
56 qq   DATA 7f,ff,4e,f9,00,00,00,00,61,28
57 Os   DATA 4a,9b,20,1b,e5,88,20,4b,d7,c0
58 YJ   DATA 22,6d,00,0c,22,06,e7,89,22,71
59 1i   DATA 18,00,50,89,4e,ae,fd,90,60,00
60 Nr   DATA ff,3e,61,06,50,8b,60,00,ff,36
61 m2   DATA 4a,85,67,02,52,86,7a,01,4e,75
62 xN   DATA 4a,9b,20,6d,00,0c,20,06,e7,88
63 yZ   DATA 22,70,08,00,50,89,20,1b,67,24
64 Eq   DATA 22,1b,e7,89,20,6d,00,0c,28,70
65 u1   DATA 18,00,50,8c,53,80,20,4c,24,49
66 gM   DATA 22,1b,d5,c1,22,12,d1,c1,24,88
67 rB   DATA 51,c8,ff,f0,60,d8,60,00,fe,f0
68 LK   DATA 48,e7,ff,fe,4b,fa,01,80,41,fa
69 vx   DATA ff,f2,22,50,d3,c9,d3,c9,58,89
70 Lo   DATA 2b,49,00,14,2b,7c,00,07,00,00
71 a9   DATA 00,18,61,0c,4e,b9,00,00,00,00
72 AC   DATA 4c,df,7f,ff,4e,75,20,6d,00,14
73 4M   DATA 2a,18,22,6d,00,18,d3,c5,d1,d0
74 hC   DATA 91,fc,00,00,00,04,4a,60,6a,02
75 mR   DATA 53,88,10,20,e3,08,66,04,10,20
76 Y6   DATA e3,10,64,5a,42,41,e3,08,66,04
77 qh   DATA 10,20,e3,10,64,42,47,fa,00,38
78 OO   DATA 76,03,42,41,14,33,30,00,48,82
79 Xp   DATA 78,ff,e5,6c,46,44,53,42,e3,08
80 jn   DATA 66,04,10,20,e3,10,e3,51,51,ca
81 Yt   DATA ff,f4,4a,43,67,06,b8,41,56,cb
82 p2   DATA ff,da,14,33,30,04,48,82,d2,42
83 of   DATA 60,00,00,0a,0a,03,02,02,0e,07
84 Kw   DATA 04,01,13,20,33,c8,00,df,f1,80
85 jK   DATA 51,c9,ff,f6,47,fa,00,d6,26,6b
86 Vu   DATA 00,14,50,8b,b1,cb,6f,00,00,c8
87 wT   DATA 47,fa,00,3a,74,03,e3,08,66,04
88 Hb   DATA 10,20,e3,10,64,04,51,ca,ff,f4
89 t9   DATA 42,41,52,42,16,33,20,00,67,12
90 QV   DATA 48,83,53,43,e3,08,66,04,10,20
91 KH   DATA e3,10,e3,51,51,cb,ff,f4,16,33
92 Ob   DATA 20,05,48,83,d2,43,60,00,00,0c
93 b5   DATA 0a,02,01,00,00,0a,06,04,03,02
94 wk   DATA 0c,41,00,02,67,42,47,fa,00,34
95 h5   DATA 76,01,e3,08,66,04,10,20,e3,10
96 oC   DATA 64,04,51,cb,ff,f4,52,43,42,42
97 CS   DATA 18,33,30,00,48,84,e3,08,66,04
98 aj   DATA 10,20,e3,10,e3,52,51,cc,ff,f4
99 OB   DATA e3,4b,d4,73,30,04,60,00,00,32
100 99  DATA 0b,04,07,00,01,20,00,00,00,20
101 Qf  DATA 00,00,42,42,76,05,42,44,e3,08
102 6m  DATA 66,04,10,20,e3,10,64,04,76,08
103 9h  DATA 78,40,e3,08,66,04,10,20,e3,10
104 uj  DATA e3,52,51,cb,ff,f4,d4,44,45,f1
105 KJ  DATA 20,00,48,c1,d5,c1,53,41,13,22
106 ge  DATA 51,c9,ff,fc,60,00,fe,c8,4e,75
107 Jz  DATA 00,00,00,00,00,00,00,00,00,00
108 kO  DATA 00,00,00,00,00,00,00,00,00,00
109 11  DATA 00,00,00,00,00,00,00,00,00,00
110 EC  DATA 00,00,a0,5a,00,00,03,ea,00,00
111 X3  DATA 08,38,00,00,36,c4,00,00,20,de
112 74  DATA b2,f3,1a,cc,09,ee,b8,61,53,e0
113 ZQ  DATA 49,33,66,e9,5a,11,0d,81,c4,54
114 1x  DATA b1,93,c9,ce,51,da,28,40,db,2d
115 N6  DATA 1a,4a,ac,00,ac,66,1a,03,7f,30
116 OD  DATA 87,ec,00,5c,a4,4e,8c,a6,2e,02
117 QU  DATA 4c,ec,82,65,2c,e6,49,fa,2d,b2
118 q8  DATA 70,13,18,dd,fa,45,e9,2a,e1,9c
119 SK  DATA 97,80,fe,37,a9,98,5c,3a,70,93
120 gq  DATA 8e,d4,0a,c0,db,82,25,25,e0,da
121 XO  DATA f5,32,b0,ee,57,08,1c,c0,53,f1
122 p0  DATA d9,0a,2d,4d,91,c2,02,f6,e5,35
123 HC  DATA ea,39,7b,2a,30,64,11,00,b0,02
124 Kg  DATA 06,bc,5d,b7,28,00,3c,0e,d4,66
125 n5  DATA fb,e9,29,f4,20,3a,35,22,ed,87
126 qI  DATA 06,4a,31,40,00,02,48,40,d0,33
127 Hu  DATA 29,72,62,6e,d6,33,3a,71,ca,31
128 OJ  DATA b6,63,29,92,09,5e,d6,93,10,d9
129 gI  DATA 7d,da,fc,ab,2a,16,e8,4e,97,be
130 14  DATA bb,05,88,14,ca,25,c2,bc,dc,9d
131 fA  DATA df,25,b0,b2,04,01,ac,0d,ce,68
132 TU  DATA ea,8a,f6,82,31,64,cb,c0,37,33
133 iE  DATA d6,70,ff,42,81,f7,20,33,de,7c
134 dH  DATA 44,d5,af,8e,70,02,cd,30,01,40
135 8P  DATA 33,2f,cc,6d,dc,31,38,97,33,7a
136 d9  DATA 69,94,52,9d,7a,76,6c,d4,a9,8c
137 GC  DATA 7a,4b,07,29,fd,ac,f8,57,c4,46
138 6C  DATA 20,2c,8f,cd,30,72,f2,05,be,4b
139 9a  DATA fa,02,fd,25,04,d7,2c,2b,98,b2
140 VO  DATA 1f,58,00,07,df,ae,1a,08,6b,a8
141 xx  DATA 0b,30,97,f7,6b,2d,e8,82,bc,17
142 1R  DATA d9,4d,30,f6,5c,a6,c3,22,10,cc
143 Zb  DATA 25,10,c3,67,7b,5b,e6,6c,40,34
144 8h  DATA 14,b0,51,80,01,94,e7,47,c2,99
145 fH  DATA 19,04,30,2c,4b,50,d6,c7,c1,ea
146 eu  DATA 93,02,3f,d3,30,96,15,af,f6,22
147 Xj  DATA 91,71,6c,ca,ff,b2,b2,05,30,82
148 EI  DATA 9c,85,02,24,b0,3c,13,01,78,1e
149 a6  DATA 30,6a,dc,06,d5,6f,62,6b,32,e4
150 ud  DATA 4a,69,20,67,6a,70,56,fe,3e,97
151 XT  DATA 05,b0,79,18,9a,6b,0a,6b,39,7c
152 AO  DATA 22,39,d4,fe,86,97,95,70,b0,8d
153 8R  DATA ba,84,73,dd,a4,de,6e,00,78,20
154 tx  DATA 50,20,68,00,0e,d1,fa,71,65,68
155 pa  DATA 96,30,0c,22,90,c0,87,78,ba,20
156 o5  DATA 89,9b,2f,fc,22,88,c5,5d,30,02
157 s6  DATA c4,45,80,24,a9,db,45,40,40,72
158 U1  DATA 23,4a,35,89,49,f9,78,86,00,29
159 cm  DATA 0c,ec,00,80,42,6c,00,88,20,1f
160 aE  DATA 4d,0b,56,43,fa,0a,96,20,3e,0d
161 06  DATA 6c,23,c9,6e,00,8c,03,69,6e,74
162 hw  DATA 75,69,74,69,6f,6e,2e,6c,69,62
163 qN  DATA 72,61,0e,c0,7f,3b,59,ca,23,fa
164 ny  DATA 0b,bb,25,ac,6c,20,70,20,ba,51
165 Lz  DATA 16,e0,2f,9a,23,ee,21,27,91,9c
166 hu  DATA 80,22,7a,2f,92,23,dc,5e,5d,10
167 M7  DATA fe,a6,7d,9d,75,b7,47,ce,4e,19
168 SL  DATA 01,66,f7,b9,ff,5e,cf,6a,7c,fe
169 09  DATA e0,e1,7c,38,10,de,40,06,38,38
170 NU  DATA c0,40,06,a0,a8,25,1a,fe,61,0a
171 zT  DATA da,43,9a,d5,9b,fe,82,99,b7,76
172 rK  DATA 5a,c4,40,e9,73,00,26,61,0e,76
173 BG  DATA 05,95,7b,ab,b6,42,18,65,7e,98
174 tO  DATA 23,58,42,83,24,08,52,43,4a,18
175 ut  DATA 66,fa,f9,a3,f3,4c,79,4e,6f,74
176 hq  DATA 20,65,6e,6f,75,67,68,20,6d,65
177 Oq  DATA 6d,6f,72,79,20,66,72,65,65,07
178 Si  DATA fc,2f,92,0a,15,63,6e,4c,16,7d
179 gr  DATA 6b,d9,12,26,73,5f,b4,96,22,6d
180 u8  DATA 32,25,37,9b,31,4b,c1,8a,97,cc
181 cs  DATA 86,4d,e6,b9,26,73,43,b6,96,ca
182 Lz  DATA 6d,33,cd,b7,9b,fa,9b,23,80,d7
183 Jk  DATA d4,b6,31,19,97,3d,cd,e6,4a,66
184 35  DATA 73,4d,00,9d,39,99,a0,5b,4d,d4
185 UZ  DATA 64,0e,06,63,b2,18,5a,6a,b6,cd
186 Ln  DATA 6d,dc,66,7a,2e,1c,c6,06,03,f0
187 Ot  DATA ea,e0,03,d1,de,6c,8e,c9,e4,60
188 90  DATA c6,5a,64,83,d4,1a,5d,d4,a0,e2
189 u1  DATA 8a,c2,0a,bc,6c,8e,c9,11,2a,01
190 QY  DATA 14,71,be,81,0b,78,4c,a6,04,0b
191 Lz  DATA f0,e5,6a,c1,02,3c,39,98,c9,2d
192 JG  DATA 30,f0,90,52,91,38,0a,ec,80,82
193 XD  DATA 16,00,68,63,72,69,fe,f2,d1,01
194 ne  DATA 3f,26,b9,a9,6c,f3,16,0d,b0,9f
195 wi  DATA 2c,fd,d0,2c,b5,25,2c,58,97,17
196 AA  DATA 01,b4,0a,0c,40,70,16,98,c8,ae
197 xt  DATA 5b,55,24,b0,0f,26,b4,ab,a4,34
198 JA  DATA 3e,ce,b5,84,c8,4e,28,75,44,03
199 ct  DATA eb,01,80,07,2c,12,65,1e,d1,9c
200 tw  DATA b8,ca,52,5f,45,30,a0,9b,75,d8
201 Ah  DATA 52,f5,ec,b2,10,2c,7a,2c,80,d8
202 z5  DATA a7,e0,20,32,6b,cc,02,3a,fe,aa
203 Vh  DATA f0,b2,0c,75,f0,ca,d4,a7,10,20
204 G7  DATA 15,fd,40,6c,9b,b8,ad,93,20,15
205 1g  DATA 08,22,2d,00,04,e0,bb,43,fa,2b
206 z6  DATA a6,e5,a3,b6,0a,db,ec,8b,10,b3
207 Da  DATA a9,b7,3a,26,d6,8d,cb,9a,da,36
208 4F  DATA 26,e3,36,b2,d9,42,dc,6c,ae,a9
209 xC  DATA 6d,63,32,2e,5b,a6,6d,7b,16,2d
210 NR  DATA ae,a9,6d,63,32,2e,55,9b,cd,62
211 nN  DATA cd,e6,9a,da,a3,a5,c4,fa,2d,65
212 qh  DATA 6a,32,72,37,28,39,9a,0d,d4,3a
213 JX  DATA a3,f5,43,4b,3f,56,07,86,57,1c
214 nv  DATA 9a,47,20,4d,4f,bc,81,04,22,07
215 Lv  DATA 4f,4b,dc,37,09,dc,12,6b,6e,c5
216 sc  DATA 4e,09,ab,42,58,52,2b,0a,66,72
217 Qr  DATA 4f,00,5e,20,0b,52,41,eb,a0,01
218 Ue  DATA 0d,4a,55,4d,50,20,54,4f,c8,df
219 aW  DATA c5,61,ff,21,0d,f1,96,45,54,ff
220 H3  DATA 01,0f,53,41,56,21,7e,43,12,33
221 xr  DATA 54,52,4f,4c,e4,40,2e,2e,2e,3a
222 1S  DATA 7d,83,8c,39,1f,13,54,4c,4f,43
223 vg  DATA 41,42,4c,45,7c,20,03,20,20,60
224 Zn  DATA 14,13,44,41,54,41,20,2b,20,49
225 f9  DATA 4e,46,4f,20,fc,2b,14,90,f0,0f
226 f3  DATA 41,44,d1,03,53,53,20,52,45,43
227 3d  DATA 52,55,4e,43,48,45,52,20,fe,03
228 C1  DATA 78,4f,4e,66,a7,4f,46,46,8d,07
229 Xz  DATA 2a,81,0c,30,30,80,4f,e0,ff,dd
230 yY  DATA 9a,71,a1,eb,9d,ff,ec,e5,19,3a
231 W8  DATA c7,ff,cb,e5,68,01,e2,6f,88,57
232 St  DATA 3d,22,c4,ea,b1,36,d8,d5,18,c2
233 We  DATA 20,7a,2a,68,31,6b,1f,44,4a,82
234 30  DATA 07,a6,08,9a,67,88,e7,03,67,57
235 ZI  DATA ff,98,c9,1d,b2,f2,72,ff,75,79
236 eZ  DATA 24,a8,30,30,00,18,8c,cf,cf,65
237 tk  DATA b6,85,24,7a,c0,18,78,3a,37,62
238 Of  DATA 52,50,e8,8e,03,66,ca,03,42,64
239 LA  DATA 02,42,59,4a,03,66,ff,ff,4c,f2
240 ME  DATA bf,dd,08,b6,db,0c,ad,91,fe,f9
241 gu  DATA f9,46,a8,c5,b6,5a,18,3c,cb,f2
242 fD  DATA 30,ff,1b,73,50,d8,a3,a3,01,f4
243 sH  DATA 2a,6f,c9,07,6d,e0,ea,95,00,8a
244 Gu  DATA de,27,d0,aa,18,0a,c1,5a,12,16
245 n6  DATA 74,fe,e3,a2,5d,e8,5a,2e,14,b8
246 a9  DATA 49,b5,fe,6e,c3,16,ac,72,69,20
247 Lk  DATA 65,a6,0c,40,2e,2f,62,e1,7c,00
248 dL  DATA 0c,2d,87,ae,69,c7,c4,28,5a,cc
249 UZ  DATA 84,30,04,1e,80,fe,9d,e7,26,09
250 B1  DATA bb,18,01,72,98,9f,e4,09,36,5e
251 pv  DATA 95,09,00,fa,6f,43,11,ac,f7,bb
252 5o  DATA 17,88,dc,64,94,97,fe,61,80,a5
253 6B  DATA 13,66,30,3f,c9,bd,fe,4f,90,6b
254 wc  DATA 2a,e9,b0,80,65,59,0a,73,be,9e
255 AW  DATA 51,8a,5a,1a,19,bc,28,95,48,02
256 x9  DATA 16,c1,fe,27,72,97,0b,dd,30,fc
257 Ur  DATA ed,97,0b,dc,a6,e8,6e,09,e0,92
258 4t  DATA 9d,d8,a2,1a,17,12,ec,45,11,1e
259 rt  DATA e7,31,eb,1a,a6,f0,05,83,13,1c
260 RC  DATA f3,d9,56,c7,b4,28,15,04,ff,18
261 ym  DATA 9e,c3,c4,b4,5f,aa,72,77,63,0a
262 AG  DATA 1d,a3,03,01,b8,5e,6e,a1,a1,43
263 Ce  DATA 15,ad,82,15,01,e7,00,76,7c,e4
264 38  DATA 15,eb,91,0d,af,4d,62,1d,f2,73
265 Te  DATA 43,f9,74,92,10,d6,f9,16,8c,59
266 2C  DATA 2e,f2,08,45,9b,15,98,25,fd,65
267 31  DATA 01,1b,f9,12,6a,cc,47,be,b2,12
268 tC  DATA e9,2d,fd,53,2c,53,08,61,7f,6d
269 ru  DATA c5,ca,a0,0c,f2,7b,de,62,98,fc
270 1j  DATA 96,54,9a,9a,fd,3e,19,f1,66,2c
271 gM  DATA 34,bb,9b,80,75,4d,96,b6,f8,c4
272 59  DATA 01,9a,4f,ba,fd,1b,ec,25,06,f8
273 bZ  DATA 46,cd,fd,0f,72,26,57,55,fd,05
```

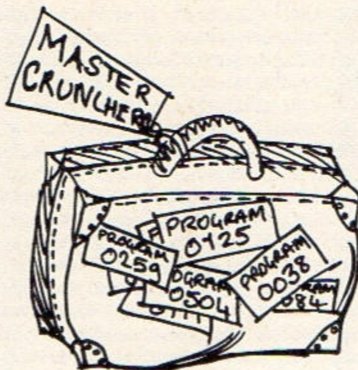





```
274 le  DATA 31,6f,52,f5,9a,07,9b,4c,6d,8b
275 nC  DATA ea,e1,16,55,03,e7,5b,38,40,03
276 k9  DATA e4,09,0e,68,85,b4,fc,d1,dc,95
277 nb  DATA 06,d4,96,8f,66,3e,66,9e,bf,2a
278 Nz  DATA fe,14,07,4b,d0,69,15,d9,41,fc
279 uv  DATA a9,5a,de,6a,03,fc,a1,20,3a,fd
280 4p  DATA fa,b7,af,e6,0d,12,e0,14,6c,1f
281 WN  DATA fc,8b,b9,a2,60,2c,6b,4f,81,d5
282 CW  DATA 0b,67,43,0b,7e,07,fb,73,2d,51
283 PM  DATA cf,b6,fd,67,fc,6b,e6,a5,82,fc
284 i1  DATA 63,c9,f1,bd,0d,e2,48,56,45,bc
285 W5  DATA fd,8e,d0,fc,70,b2,03,16,79,54
286 G2  DATA ea,15,ac,4b,fb,f6,5b,3e,58,a1
287 U2  DATA 70,37,fc,42,b3,00,fd,86,e6,a3
288 J3  DATA df,f1,9c,42,fd,78,20,db,11,bd
289 i4  DATA 11,1f,3c,76,54,21,fd,74,33,98
290 Oe  DATA cc,fc,1e,0c,a8,fc,1a,33,4b,c9
291 Wc  DATA fe,60,54,4e,a1,e5,54,f9,b3,2d
292 zM  DATA be,fb,02,da,0e,6c,4c,e8,af,51
293 4q  DATA 14,9a,0f,56,cb,c7,1a,c2,89,c5
294 Jn  DATA 3a,5a,63,fd,32,38,1b,c3,fb,da
295 hG  DATA d9,18,fd,98,ce,45,cd,c5,fc,20
296 Ug  DATA 72,51,a1,b1,fe,0e,5c,94,db,90
297 75  DATA 70,c9,45,09,2a,fc,48,60,a7,6e
298 Jh  DATA 91,f9,26,4b,fa,f8,2a,ed,f7,72
299 Y7  DATA 52,fd,a4,e6,35,a8,e9,14,5f,57
300 xP  DATA 76,00,82,39,38,ab,d6,d4,96,59
301 2o  DATA d9,7c,60,dc,58,09,f8,5b,55,0a
302 OD  DATA 0c,31,0b,f4,3e,ce,10,b3,f7,99
303 ud  DATA 85,56,bf,55,46,b5,2a,56,1f,77
304 4W  DATA 56,60,55,0a,0e,a9,56,aa,55,63
305 8W  DATA bd,85,05,ba,00,f7,37,27,e5,ea
306 9z  DATA f8,b3,5b,be,14,73,0c,95,18,76
307 zF  DATA ed,09,83,4d,1d,2b,6c,6d,56,a8
308 FZ  DATA 0e,6a,2c,c7,c3,22,6d,51,a9,b2
309 q0  DATA 11,5c,5c,d4,21,fe,e4,2c,17,75
310 wN  DATA 08,05,38,cb,d5,b0,0e,61,03,94
311 HE  DATA 60,d0,3d,08,5e,56,3c,99,ff,48
312 nL  DATA b9,5f,5e,5b,6b,d5,01,bf,41,53
313 11  DATA 55,0a,08,7c,66,4e,61,5b,55,09
314 wn  DATA 7c,76,14,9c,83,49,11,f8,9e,00
315 7G  DATA 3e,dc,29,be,4b,16,f8,2a,f3,c5
316 rk  DATA 89,a9,f0,6c,9b,0b,db,44,6d,b5
317 Rc  DATA ba,cf,34,95,50,d8,fe,32,2a,08
318 ZK  DATA 10,15,8d,df,f0,0a,74,e5,29,3a
319 Y1  DATA 25,24,94,00,e3,c1,e7,7b,32,8c
320 1r  DATA c3,14,95,51,45,ba,ef,12,14,80
321 R2  DATA 30,10,32,11,fc,10,2a,70,aa,7b
322 2c  DATA 00,0f,67,02,72,2d,ff,5d,8b,00
323 x1  DATA 30,3c,01,35,0c,02,0f,f5,6d,04
324 gF  DATA 32,e1,d0,f5,30,80,32,81,06,eb
325 V8  DATA 51,80,3c,28,20,7a,28,14,20,0f
326 Kr  DATA 10,81,23,ee,ee,02,2a,08,48,13
327 da  DATA 33,2b,c3,8e,e2,48,e5,c2,01,06
328 gx  DATA 5d,16,0b,30,70,02,08,97,0d,08
329 qI  DATA 67,06,08,e8,f8,55,03,58,ff,f6
330 Jp  DATA f9,16,00,53,e2,a2,6a,ba,80,55
331 bs  DATA 09,89,b6,75,2c,24,74,37,77,64
332 mG  DATA 95,b2,c6,3d,c0,6a,89,24,6e,e5
333 3S  DATA 50,10,22,10,d9,06,cf,d9,84,51
334 MS  DATA 89,2c,5f,31,73,18,7a,8a,af,82
335 U0  DATA 14,4e,1b,80,17,77,2b,21,c8,56
336 Gg  DATA 31,1a,24,38,67,e1,98,85,da,d1
337 bV  DATA 2f,9a,ed,cd,bd,0b,a2,0d,aa,e5
338 eK  DATA f0,b2,59,73,ee,1c,91,f5,dc,00
339 Fk  DATA 8a,3c,51,77,00,b6,33,d4,8d,1b
340 z1  DATA ab,15,75,b7,61,9b,ba,35,6c,53
341 AS  DATA b7,86,6d,ea,d6,b0,4d,dd,09,a0
342 KT  DATA f2,c4,dd,6c,f2,db,0b,08,38,f3
343 mC  DATA db,0b,40,0c,04,b8,97,c2,5c,1a
344 ne  DATA b0,41,29,24,56,24,1a,2d,94,66
345 di  DATA 8b,61,22,bd,84,aa,a3,6b,1a,38
346 kp  DATA b4,a9,d9,7a,06,64,73,eb,26,ac
347 t0  DATA 2e,b3,d8,00,c6,cc,05,58,d4,77
348 Zw  DATA 23,7c,8c,17,36,b1,61,36,27,23
349 Oh  DATA 70,cb,06,34,74,8a,95,c0,fe,ea
350 j1  DATA 66,26,dc,7e,01,99,87,ee,7e,02
351 14  DATA f6,e8,7e,50,c6,e1,c5,34,4d,d1
352 Ut  DATA 8e,83,46,0a,18,72,53,29,b1,26
353 05  DATA 62,6c,26,52,34,8e,9e,96,b5,62
354 bP  DATA 04,d6,a2,ac,16,35,a9,08,8e,cb
355 Th  DATA 61,b7,a8,ae,c0,ca,6a,25,f2,4a
356 HW  DATA 91,21,f7,22,51,60,05,24,49,43
357 VA  DATA e9,79,fe,a8,ad,ac,6a,04,42,19
358 Lg  DATA 60,e2,79,6d,01,70,30,ee,3a,04
359 f7  DATA ed,11,d5,c7,26,06,32,a2,48,e7
360 qF  DATA 00,22,5c,24,70,40,66,66,8c,01
361 OY  DATA 4c,df,44,00,78,5f,04,70,26,a7
362 A6  DATA e6,24,80,42,80,60,f8,f6,45,dd
363 uM  DATA c4,4a,99,a2,86,62,00,24,b9,b4
364 8X  DATA a8,45,1c,2d,6a,ac,57,b6,43,34
365 pg  DATA dc,b0,4b,0c,d6,01,59,55,b2,10
366 4E  DATA d6,2c,20,49,70,18,23,ef,98,e5
367 Kq  DATA c8,21,3d,84,52,25,a0,9c,58,43
368 f2  DATA 23,7a,0e,15,b1,9d,fc,37,4a,53
369 1R  DATA 40,60,ee,ce,5b,dd,1e,18,cc,45
370 gu  DATA 3c,7c,16,12,d8,f9,03,06,5b,1b
371 BK  DATA a8,8d,52,46,4a,07,d7,c5,0c,46
372 8m  DATA 00,06,65,22,d6,5b,59,5b,50,0b
373 MW  DATA ed,49,dc,a0,52,0e,5d,5d,65,3f
374 rF  DATA 9b,20,51,ce,ff,fa,f9,a9,40,52
375 3Q  DATA 3c,07,0b,51,ff,fc,66,ce,83,3a
376 Zg  DATA 35,2d,57,92,25,ac,aa,f3,08,00
377 ip  DATA eb,14,ff,e6,2c,3f,db,81,5c,f7
378 gB  DATA 28,5e,2e,4a,9e,7e,66,cb,f8,12
379 wf  DATA 56,43,13,d2,0b,d5,cc,d5,fb,92
380 q5  DATA f2,8f,fc,76,85,f2,06,b6,d1,1a
381 bL  DATA 05,bc,b9,15,d6,58,6e,34,b6,16
382 1W  DATA 21,78,2c,2d,22,32,14,54,21,15
383 mB  DATA 26,33,11,cc,17,ca,8d,81,0b,17
384 9s  DATA aa,20,ba,f8,14,d8,9e,7e,14,f0
385 Nn  DATA 6c,14,06,2d,9e,a4,ea,10,9c,a5
386 7u  DATA 82,fb,2e,73,95,98,b6,8e,ab,ed
387 um  DATA 52,74,49,01,5e,ef,02,ad,7f,b2
388 LC  DATA c2,15,6e,1c,0e,c4,ab,01,48,18
389 zG  DATA 21,7c,67,84,18,45,fa,f6,6c,f7
390 k1  DATA a5,0c,5c,87,3a,cd,31,7a,f6,5e
391 Sp  DATA f5,21,24,08,76,08,39,93,45,36
392 8P  DATA 75,84,de,12,20,e2,88,41,e0,15
393 zS  DATA 2a,4c,15,22,20,fe,90,b4,2b,82
394 bM  DATA fb,1e,d9,15,c3,67,22,5d,ed,c5
395 E8  DATA 1e,ee,6e,bd,2a,dc,d0,f8,5b,e7
396 MZ  DATA 60,d9,70,f4,90,5b,20,98,cc,02
397 su  DATA 5a,2d,96,d5,99,aa,14,e4,58,5c
398 vJ  DATA 14,dc,0c,00,b3,0c,6e,7a,a2,d8
399 dy  DATA fd,ec,ea,3a,ac,27,20,62,5a,56
400 RO  DATA 83,43,82,02,d6,22,7a,20,f3,e3
401 WG  DATA b8,c8,61,02,ca,66,72,98,49,63
402 xu  DATA 4e,30,6b,32,6a,bc,ae,25,1d,58
403 5I  DATA 63,77,0e,b2,19,c2,9a,50,c2,fa
404 z6  DATA 8c,cc,63,b0,e2,fe,cd,4a,da,23
405 eJ  DATA 88,e5,fb,59,0a,a8,61,a0,7d,66
406 Fw  DATA b8,2b,0b,44,01,da,31,03,fa,06
407 L1  DATA 77,d8,fe,55,c0,02,72,3c,45,d1
408 iI  DATA 54,ee,42,81,e1,8c,b1,19,6d,d8
409 U3  DATA 1f,da,04,b6,af,6e,4d,db,28,0d
410 aW  DATA d2,b5,d9,d0,52,06,80,8c,95,6b
411 JJ  DATA 31,ca,8c,48,84,a6,ac,1f,1f,a8
412 v6  DATA 66,14,14,1f,73,8b,1f,9e,96,6d
413 F8  DATA a7,ab,c4,21,bc,22,08,59,c7,91
414 TX  DATA 03,ed,99,b7,e2,da,21,eb,51,96
415 vw  DATA 22,e8,05,4f,d8,22,00,24,db,05
416 i1  DATA 6e,26,3a,1f,72,f8,a7,d6,de,6a
417 EA  DATA 50,01,99,13,c0,99,66,22,c6,fc
418 af  DATA dc,43,0f,50,a8,8f,50,41,43,4b
419 CJ  DATA 3c,a3,56,0e,2e,d4,4e,f9,01,6a
420 eR  DATA 79,98,4a,a8,01,6c,51,7e,e5,96
421 oX  DATA 14,cd,ee,ae,b0,50,15,1f,26,41
422 HH  DATA e8,03,90,2f,08,e1,97,fd,44,e0
423 PI  DATA 40,02,20,5f,f6,27,ce,2b,48,5b
424 Y1  DATA 59,f6,a5,3c,1c,56,02,79,d4,54
425 cK  DATA 22,c2,90,5a,1f,9a,e9,3a,dc,bf
426 tx  DATA fa,20,80,2b,58,97,18,90,06,15
427 37  DATA e6,f5,1b,76,f2,62,56,47,69,92
428 OW  DATA 0c,c4,9c,3a,0c,a8,56,31,2e,32
429 43  DATA 40,5f,46,66,ee,c6,1c,82,f2,44
430 1F  DATA 53,01,1e,ba,c6,19,71,2b,b9,1c
431 17  DATA 99,e5,af,f6,67,0e,a8,19,f9,c4
432 NM  DATA 30,f4,16,4a,52,66,5a,fe,92,01
433 mA  DATA 1e,96,35,00,40,66,d5,e5,b3,1c
434 sY  DATA c9,2c,0d,10,c0,08,0e,86,5c,ec
435 f0  DATA 61,5e,b5,4b,6c,a8,21,f6,01,23
436 KZ  DATA c9,0c,fc,98,42,50,c1,8f,5a,af
437 c1  DATA 00,e8,65,12,2e,f8,ca,9c,2e,55
438 qk  DATA 29,9a,59,c1,4b,fa,16,0c,22,cc
439 24  DATA b7,18,ce,2a,1a,2a,42,b3,70,12
440 Cw  DATA 58,fd,a7,7b,6e,99,b1,0c,0a,e3
441 73  DATA c8,2c,16,1b,96,9c,1b,aa,60,b8
442 UI  DATA d8,41,b5,c5,0a,a2,f2,44,c6,ac
443 2D  DATA db,b2,66,4e,79,15,1e,16,87,27
444 09  DATA e4,14,56,68,c3,b8,f3,63,7c,7a
445 tl  DATA 41,a7,3a,30,3a,f3,62,e3,48,30
446 eW  DATA 33,e6,a7,11,30,f6,48,c0,d2,80
447 03  DATA cb,4b,f8,26,81,cc,53,3f,65,31
448 v0  DATA 21,11,16,72,27,86,b4,5f,29,ad
449 9F  DATA a0,44,03,a0,00,08,01,f2,2d,bf
450 yq  DATA 0c,61,1d,c2,e6,31,b2,58,8f,b6
451 Wb  DATA d1,f2,1f,f3,07,2c,e7,b0,d1,6d
452 hw  DATA 3c,15,d7,1b,0b,4e,67,c6,f2,f1
453 uC  DATA 17,33,cd,ed,38,c1,20,7a,1d,78
454 U1  DATA 0c,98,78,bf,ea,f3,66,60,db,43
455 U2  DATA 5c,25,4a,98,66,56,22,18,52,81
456 Tc  DATA b2,80,66,4e,11,7f,1b,ec,e9,a8
457 BG  DATA c5,ca,f6,ea,a8,c5,c0,f6,eb,a8
458 JB  DATA c5,c2,f6,ec,a8,c5,be,f6,f2,a8
459 fW  DATA 9c,31,00,c4,73,f1,51,8b,14,5a
460 Z1  DATA ef,d4,a2,48,0c,68,03,f0,fc,17
461 CP  DATA 40,70,2c,c7,c4,a2,a0,03,c1,0c
462 pb  DATA 1d,64,c9,80,d0,fa,ed,10,c0,ea
463 bW  DATA 3b,61,81,78,2a,48,22,48,e7,63
464 yY  DATA 0e,02,80,74,df,00,1d,0c,4c,60
465 07  DATA ee,20,a6,11,b8,90,88,24,48,95
466 OT  DATA c9,fc,a4,b2,16,86,73,55,1c,a2
467 ZF  DATA 92,8a,23,c4,ef,f0,32,82,b2,9a
468 3x  DATA 8c,fd,90,20,4d,7b,30,52,9d,6c
469 Ff  DATA da,08,24,50,88,9a,b1,67,06,20
470 5g  DATA 18,bc,0d,16,ab,e5,88,d1,c0,60
471 OR  DATA f2,58,88,b1,41,7e,9d,64,6d,b1
472 Ko  DATA 04,6e,00,77,f1,22,c1,ae,b4,da
473 8f  DATA 01,78,22,08,74,fe,cb,67,ac,15
474 ZC  DATA 28,67,55,32,20,80,fb,9a,aa,57
475 kH  DATA 28,25,eb,33,68,99,a3,9a,1a,25
476 Ku  DATA 1e,6b,27,86,2c,7a,6c,2b,4d,a5
477 DJ  DATA 0e,af,f3,b1,a6,9e,f6,2d,5a,97
478 re  DATA ca,ac,1b,62,aa,69,cc,1f,60,42
479 RB  DATA 90,1e,29,bc,ab,ee,e6,01,93,ee
480 nj  DATA cb,fe,ef,58,63,65,15,8d,8b,0a
481 yl  DATA 66,70,09,f5,a3,98,6d,20,f6,d9
482 sg  DATA b2,62,01,16,54,db,c8,8e,1e,0c
483 Kl  DATA 52,40,45,75,5a,7c,f9,6a,eb,4a
484 yD  DATA 1b,f8,97,8b,36,5a,02,9a,ab,0a
485 4q  DATA 2e,00,97,c0,70,2a,1f,0a,c2,fa
486 iI  DATA 1e,fa,62,7f,b8,d3,5f,00,30,7e
487 NW  DATA 0b,f9,0c,c1,85,f3,0e,72,fa,05
488 HC  DATA 4a,fd,64,4b,03,03,7b,34,f4,6b
489 9e  DATA 04,51,17,12,68,07,51,a7,a6,5a
490 On  DATA 14,d4,a9,24,0e,65,4c,2a,8b,05
491 7N  DATA 6a,02,5c,c0,45,95,d6,03,e2,72
```

Listing. »MasterCruncher_Gen« (Fortsetzung)

```
492 Ap  DATA 0a,51,83,0b,b9,c1,0c,dc,60,0d
493 Mk  DATA 6e,30,0e,17,4d,01,52,0c,44,fe
494 HI  DATA 47,2b,b0,ff,88,7c,7e,30,94,37
495 q2  DATA 6c,60,06,79,74,42,50,68,2b,1e
496 lw  DATA 6e,4a,50,66,2a,f4,62,1d,32,b3
497 he  DATA 1b,36,96,a1,ba,04,d4,58,71,60
498 6g  DATA 06,5e,c0,4b,fa,ec,c4,51,de,ed
499 Yz  DATA ec,80,67,66,b1,00,09,40,a9,37
500 YR  DATA 6c,a7,c0,00,20,3d,22,7a,1a,24
501 JQ  DATA 72,00,32,00,ed,48,e1,49,d2,40
502 Om  DATA 41,f0,68,00,ff,05,43,f1,10,1b
503 qS  DATA 72,07,10,1a,12,80,46,00,10,80
504 xI  DATA d0,00,f8,1f,d2,54,28,1d,b3,68
505 LG  DATA ee,4c,9f,5b,30,61,3e,66,ce,80
506 xn  DATA 44,b5,28,e6,9c,6c,0c,07,1b,ce
507 zd  DATA ff,5a,67,f6,1d,ce,61,a7,1d,be
508 7a  DATA b1,f7,de,0e,4f,d0,61,02,60,d8
509 aG  DATA 91,9f,96,ea,32,80,31,57,ca,60
510 e0  DATA 14,cc,1a,80,25,0e,44,6a,46,50
511 CB  DATA 83,30,da,3a,49,e3,b4,ab,08,a6
512 8q  DATA 5a,0e,8d,f5,4d,08,6b,2d,00,dc
513 RI  DATA 99,3a,61,4c,66,2e,98,6a,01,44
514 uA  DATA 2d,90,6a,01,5d,c0,f6,88,72,01
515 ZX  DATA 34,2d,80,6a,01,2d,c1,f6,78,42
516 YI  DATA 80,61,24,f6,45,5a,70,01,61,1c
517 M8  DATA f9,a2,fc,04,6c,14,16,4a,70,03
518 fL  DATA 61,0c,e5,8b,42,3a,c3,8b,3e,67
519 p4  DATA 64,3a,1d,24,d2,f2,6a,1c,d6,4a
520 PM  DATA 41,67,06,20,50,53,41,60,f6,41
521 50  DATA e8,5f,ff,80,b4,14,01,67,22,4a
522 69  DATA 18,66,20,e7,7b,1b,8c,70,81,01
523 Ke  DATA 32,de,a2,10,a2,1b,74,30,08,08
524 WH  DATA 0a,6b,d7,19,c2,4a,19,66,fc,53
525 OJ  DATA 49,0c,29,ac,1f,6a,ac,6b,12,56
526 J8  DATA 43,2f,f2,67,04,12,d8,60,f8,42
527 G8  DATA 19,0c,bf,13,27,01,17,de,95,19
528 B7  DATA c3,7f,07,58,8f,68,00,14,79,88
529 Ln  DATA 14,d6,02,ae,24,b2,8b,9e,6c,be
530 1K  DATA a2,3e,30,1c,a0,73,b0,96,0d,88
531 Rd  DATA b6,04,36,86,42,90,2f,0e,c8,23
532 pU  DATA 1c,3a,22,52,70,3f,42,1a,09,7f
533 Hq  DATA 0c,fc,b3,39,da,67,16,2a,51,70
534 9v  DATA 40,bd,24,af,22,4d,bb,f7,a8,de
535 01  DATA 66,ea,2c,5f,8e,42,0d,fd,66,c7
536 SN  DATA 1c,4c,4a,90,e8,7a,30,07,58,f3
537 78  DATA 40,53,50,de,83,25,80,6b,1c,36
538 Dv  DATA 2c,f6,4e,c0,34,3b,30,3c,ce,02
539 d0  DATA 72,c8,fe,ab,a8,54,08,e0,f8,ad
540 dc  DATA 10,c0,db,89,00,e8,65,07,22,0d
541 Nk  DATA c6,a0,01,2e,73,07,7e,e5,c0,03
542 9K  DATA 00,ac,01,03,f7,76,fc,f4,96,53
543 oY  DATA e4,0f,36,b8,35,45,23,6e,0b,62
544 TG  DATA ed,a8,a0,2d,e2,74,45,24,db,ee
545 87  DATA 42,09,1a,3e,78,45,c0,9b,22,ec
546 6N  DATA 77,ca,61,0e,68,23,f4,48,37,35
547 hi  DATA ba,fe,0a,ec,fd,b2,10,a8,8e,de
548 TT  DATA 12,55,c1,3e,2d,7d,10,c0,10,c1
549 Tl  DATA 3e,f1,0e,24,1a,e9,12,f0,1c,58
550 Km  DATA ac,29,92,f4,12,25,6e,b7,60,9e
551 W2  DATA 2d,8d,44,24,8b,47,fa,02,70,36
552 gs  DATA bc,cd,ff,46,11,9c,16,32,4a,10
553 J5  DATA 66,0e,e5,1d,18,28,11,7c,00,3a
554 t4  DATA 7a,bf,c9,19,c3,61,94,14,6a,99
555 TQ  DATA b5,62,1d,c1,8c,14,5a,b3,8f,06
556 TG  DATA 46,98,0b,c6,99,e9,30,3c,24,9d
557 f0  DATA 82,61,f7,f1,fb,d4,96,b3,c2,0c
558 KX  DATA 90,76,39,05,65,26,20,3a,1a,b6
559 Vi  DATA 5b,80,b0,f5,4b,ac,67,f0,92,05
560 Sf  DATA ba,2c,0f,a2,52,90,6d,3d,11,20
561 oJ  DATA ba,1a,98,70,f4,01,af,63,fe,66
562 1q  DATA b9,bc,0e,e0,04,15,c6,88,66,42
563 2h  DATA 82,42,83,76,23,30,59,40,3b,65
564 3L  DATA e6,1a,d0,4c,98,00,d6,2f,17,36
565 DM  DATA 64,e2,e8,38,17,30,65,dc,8e,17
566 Tc  DATA 2c,64,d6,e3,38,17,26,65,d0,4a
567 vq  DATA bc,d3,c0,da,6e,60,30,61,68,38
568 xn  DATA 1f,dc,48,a5,a0,66,3e,f3,fc,64
569 Mg  DATA f0,d5,be,f6,65,ea,da,57,f2,64
570 2Z  DATA e4,fb,ec,65,de,61,3a,2a,af,16
571 rV  DATA e2,64,ca,b4,2d,9e,dc,65,c4,b2
572 UU  DATA c0,09,d8,64,be,b6,5c,d2,65,b8
573 2Q  DATA a5,ab,ea,de,60,02,bd,21,45,24
```




```
574 Rq  DATA 10,4a,52,67,02,61,0a,82,9f,8c
575 M9  DATA 06,42,cb,2c,92,c9,35,2c,7a,16
576 b8  DATA 94,66,27,15,d4,e1,26,f0,58,42
577 F3  DATA 28,02,2a,03,54,40,d9,fc,eb,24
578 Vv  DATA 00,26,01,98,80,9a,81,55,44,2c
579 rR  DATA da,2f,82,50,b3,05,bb,a2,32,08
580 ru  DATA 63,65,01,24,66,8c,d3,95,43,52
581 6P  DATA 66,30,29,1d,1a,98,56,82,55,64
582 qV  DATA 25,c5,00,e3,99,93,11,9a,54,30
583 AW  DATA 01,12,2e,2b,08,20,49,04,79,95
584 ma  DATA 02,17,0a,5b,00,05,9d,d2,00,5e
585 zU  DATA 00,e1,94,8f,06,a1,4e,35,00,76
586 KN  DATA 01,3d,00,83,f7,01,77,3b,37,70
587 pp  DATA 9f,43,57,6b,39,46,b6,49,19,38
588 te  DATA e9,75,14,d9,28,30,0b,f0,d8,58
589 Au  DATA 26,65,00,91,8e,ea,18,05,06,0d
590 ue  DATA 5e,68,db,68,73,00,0e,de,9e,f0
591 e2  DATA f6,b5,89,88,b0,7a,18,84,67,fa
592 Gl  DATA 3e,07,af,3b,f1,00,45,50,51,52
593 si  DATA 53,54,55,56,57,58,59,00,46,5f
594 wT  DATA 4c,4d,4e,4f,42,00,30,31,32,33
595 wK  DATA 34,35,36,37,38,39,61,41,62,42
596 96  DATA 63,43,64,44,65,45,66,46,e5,e0
597 It  DATA 7f,01,ce,11,39,8a,11,36,d4,4b
598 E1  DATA e8,f2,aa,9c,40,53,21,42,84,38
599 oD  DATA 6b,18,2c,0c,53,00,51,67,f6,42
600 Og  DATA 53,e8,bf,a6,a1,37,ac,30,64,18
601 CX  DATA 4a,05,76,f8,f0,81,9c,32,3a,18
602 bL  DATA 0c,72,13,fe,98,66,bc,a0,b1,ce
603 MM  DATA 8a,92,20,b8,60,43,51,3b,f2,96
604 Bf  DATA 41,66,28,4a,44,67,c8,53,44,91
605 SC  DATA 3f,85,64,53,13,5d,7e,53,45,96
606 Ux  DATA 82,00,2b,07,57,83,4c,ab,8c,03
607 rx  DATA 52,60,78,dc,5c,f8,fe,4d,6d,76
608 Kr  DATA 2c,6e,00,e0,09,b7,17,aa,0c,17
609 mq  DATA b4,72,20,95,ca,b8,de,e5,66,0b
610 VD  DATA b0,1f,22,17,a4,0c,13,78,8e,4e
611 J7  DATA ac,45,14,4a,53,67,18,f9,b8,2a
612 6A  DATA 42,40,10,3a,17,8c,72,15,b0,33
613 3h  DATA 10,5a,fe,21,74,14,f8,ab,d3,b8
614 ZF  DATA 06,f8,7a,56,1a,96,0d,74,d4,fd
615 Nf  DATA 0e,10,ba,17,6c,51,4f,fd,06,52
616 nh  DATA 93,e1,03,c4,01,52,44,d1,05,d8
617 YD  DATA b4,80,de,47,ec,d0,52,13,d1,cd
618 PO  DATA b2,ba,96,c8,02,2d,98,f0,c2,01
619 mV  DATA 06,1b,7c,db,99,ba,01,1a,cc,20
620 nF  DATA 22,1d,fe,5c,9a,02,b6,2a,7a,fc
621 Mc  DATA e5,31,15,26,9b,40,17,a2,4a,50
622 Rz  DATA 67,72,0c,50,e0,2f,9f,60,b5,86
623 mM  DATA c0,46,d1,35,aa,a6,00,6a,ac,04
624 DN  DATA 4a,4a,40,67,40,d0,17,94,d0,02
625 zs  DATA 80,20,7a,b6,00,14,02,24,88,11
626 f0  DATA 17,c2,d1,92,99,77,d6,a8,f5,c2
627 Np  DATA 01,b3,e5,c8,8e,0d,e7,04,b8,ab
628 J0  DATA d9,02,59,ef,fa,f1,f6,d5,44,45
629 4p  DATA fa,ed,c0,34,bc,ee,6f,8c,ca,78
630 GN  DATA 9a,d4,0d,10,d2,39,38,a9,49,18
631 EZ  DATA 2d,84,25,80,2c,60,00,ee,0c,79
632 kQ  DATA 01,e9,05,38,17,96,d0,00,11,b9
633 NM  DATA 03,4c,da,a7,ef,7a,02,06,f0,76
634 yc  DATA 80,dd,b7,80,72,06,c8,12,25,c0
635 nK  DATA 01,24,fc,94,d7,98,de,d8,0d,e0
636 Th  DATA 51,b5,20,3a,13,4e,f0,41,1f,10
637 i4  DATA fc,00,13,79,eb,98,00,fc,19,5e
638 01  DATA e8,9f,5c,6b,b6,2b,13,9d,b5,44
639 Xw  DATA d1,6a,8f,08,d5,03,0d,00,66,5c
640 VM  DATA 80,07,1d,c6,a1,10,28,10,1b,46
641 Wp  DATA 00,e2,08,65,26,f4,01,40,da,00
642 zw  DATA 4d,01,ff,9b,ea,32,80,60,2e,94
643 rz  DATA df,e6,e0,01,e9,0d,60,20,c2,43
644 eZ  DATA 67,62,1a,0c,9d,59,6c,08,96,93
645 JC  DATA c2,51,60,1c,a7,40,00,70,c1,15
646 xx  DATA be,01,a9,ad,97,cc,6a,bf,0e,19
647 ce  DATA c7,df,39,1b,7c,c0,31,b5,42,81
648 kt  DATA 7a,04,dc,68,12,18,42,82,2e,2a
649 11  DATA 05,5a,a9,ba,80,82,58,45,f8,a4
650 4m  DATA ec,1a,e8,98,71,de,82,fc,d5,45
651 b8  DATA 03,3e,ec,41,00,12,14,19,16,19
652 f0  DATA 73,1f,01,04,b2,03,66,f2,37,1f
653 Xo  DATA 4f,30,01,31,02,32,03,33,04,34
654 LL  DATA 05,35,06,36,07,37,08,38,09,3b
655 Pg  DATA c0,7f,49,30,0a,41,0a,61,f1,75
656 mA  DATA 0b,62,0c,43,0c,63,0d,44,0d,64
657 ud  DATA 0e,45,0e,65,0f,46,0f,66,1a,e0
658 cT  DATA bf,65,cd,f0,80,74,00,e9,98,16
659 cM  DATA 7e,f5,10,0f,74,ff,0c,38,09,62
660 FG  DATA 06,41,0f,30,60,04,06,03,00,37
661 es  DATA 10,c3,cd,7f,7d,e2,ab,8c,01,0f
662 kz  DATA 18,0e,ae,1a,a9,12,0e,b4,e8,ff
663 xF  DATA c2,58,b6,12,0a,93,68,02,02,25
664 mX  DATA 10,12,04,2a,80,b8,cf,e7,62,4a
665 5U  DATA 55,66,3a,22,7a,11,de,20,09,f8
666 pC  DATA 77,cc,30,6a,35,45,47,fd,96,a0
667 vE  DATA 43,5e,61,19,5a,08,81,61,28,7a
668 VM  DATA a4,61,22,68,0f,c7,61,1c,ed,a1
669 EL  DATA ea,61,16,5d,3b,d4,a2,d0,8a,10
670 9y  DATA 4c,b9,98,2a,e5,83,7e,0b,42,80
671 JR  DATA 10,19,e0,98,e3,f7,9e,31,54,56
672 IM  DATA 48,51,cf,ff,f0,58,5f,0c,41,97
673 QR  DATA 9c,00,df,f0,06,66,f6,f9,3e,52
674 tu  DATA 67,fe,b7,cc,03,cc,af,1e,43,e8
675 4a  DATA 28,6e,3a,b8,a1,00,40,67,48,fe
676 IS  DATA a6,d0,b5,42,c1,03,40,c0,4b,93
677 zr  DATA ea,eb,52,49,24,5e,b4,a3,68,47
678 qM  DATA 70,e1,c8,65,90,1b,13,e6,a1,70
679 54  DATA 00,f0,84,60,d4,30,73,28,48,45
680 hX  DATA e1,46,60,24,11,bb,34,72,70,e1
681 Zt  DATA 1c,12,1c,48,80,34,01,e1,49,ed
682 g0  DATA 4a,d2,42,d2,40,00,f8,0f,10,2e
683 pD  DATA 3c,56,18,9c,13,b7,14,1c,67,3c
684 8w  DATA 0c,02,3d,e3,67,da,94,2a,bc,75
685 pQ  DATA e5,1a,28,34,99,0f,6d,e6,4a,26
686 EG  DATA 6a,7d,57,47,f3,20,00,42,43,78
687 Gi  DATA 07,1a,13,46,05,11,85,fe,4d,06
688 dP  DATA 43,00,28,d6,ea,d7,77,4b,63,ee
689 Rf  DATA 52,48,6c,2f,c6,c0,5a,61,66,37
690 zY  DATA 43,74,cb,a3,4e,ab,b6,33,22,5b
691 UK  DATA 92,54,52,20,3a,10,52,90,81,90
692 v0  DATA ba,10,48,84,fd,47,2e,72,f9,91
693 Mm  DATA fe,02,2c,0d,22,c3,11,94,3b,37
694 1M  DATA 0b,43,28,2a,b9,80,10,36,2a,8b
695 OX  DATA fe,9e,3a,bc,b5,8d,70,65,3b,ad
696 1y  DATA 31,f6,bb,5b,dc,54,44,0c,f9,8c
697 C8  DATA fe,88,c0,08,13,7c,42,55,ce,ef
698 re  DATA 8c,c2,43,f4,0d,f2,b6,c1,06,24
699 2K  DATA 42,de,36,d8,e8,06,e6,5b,02,08
700 nd  DATA 0f,e4,42,95,6e,f1,aa,4b,fa,e1
701 p8  DATA b0,d0,d3,e2,ec,70,00,38,b5,14
702 7V  DATA c2,b5,be,a8,45,b2,2d,c2,aa,f9
703 jM  DATA 0f,b6,b8,81,c6,d6,c5,e2,4c,d0
704 U2  DATA b6,ba,45,f0,08,00,22,78,07,de
705 5H  DATA 50,89,42,45,f1,80,c3,ff,20,08
706 9G  DATA a3,c6,1f,c7,bf,e0,31,8a,98,6a
707 4V  DATA 10,10,33,fc,89,bf,84,99,72,01
708 1E  DATA 24,3a,e6,2c,b0,30,0d,7e,66,36
709 cZ  DATA 48,e7,80,f0,42,40,43,85,fc,0f
710 UF  DATA 26,49,b3,08,66,10,b3,ca,62,0c
711 IG  DATA b1,cb,62,08,52,40,0c,e0,ff,4d
712 sc  DATA ff,66,ec,b0,ac,33,82,49,71,33
713 49  DATA 59,08,24,ba,33,db,3a,00,24,e2
714 1B  DATA 5e,59,7d,52,41,47,f0,10,00,b7
715 n0  DATA ca,57,ca,ff,ba,e2,3f,62,32,41
716 gI  DATA 86,5c,90,41,e9,a1,01,67,32,bd
717 Lv  DATA 52,0c,b7,41,68,5d,a2,6a,10,57
718 BY  DATA 63,1a,b5,41,56,0d,ca,6a,00,f2
```

```
719 7k DATA 7a,00,1b,56,a4,41,f0,c6,fc,b0
720 21 DATA 16,60,29,47,ea,ff,ff,b7,c8,b9
721 oj DATA ef,89,4a,12,d8,52,45,e5,af,60
722 gJ DATA 2a,0d,c7,fb,9a,41,20,09,e2,88
723 Ns DATA 64,06,52,49,f9,53,65,0f,32,c1
724 G8 DATA 20,9d,64,7a,ba,00,f4,93,c8,21
725 mt DATA 49,e4,cb,ce,ee,7c,ac,22,b3,2a
726 H3 DATA 76,05,c1,96,a6,ba,71,39,f6,9a
727 36 DATA 30,77,d0,02,06,48,82,6d,03,e5
728 vG DATA a9,82,45,42,88,af,30,07,23,77
729 BT DATA a2,50,e5,05,08,0f,da,a6,93,23
730 1e DATA 02,05,03,08,0a,12,dc,6f,cf,2c
731 A7 DATA 76,02,b6,38,83,db,b3,e2,4b,5c
732 st DATA 44,55,b4,4c,e3,4b,36,33,30,06
733 WQ DATA 38,03,e0,4b,a4,5f,44,e7,69,6c
734 R9 DATA 02,44,00,0f,22,2f,a0,07,16,b2
735 6x DATA 01,20,06,06,09,08,0c,0d,f8,f1
736 uv DATA 02,3f,62,d5,20,44,32,00,ea,1d
737 90 DATA 3f,6f,0c,04,41,5a,1e,66,78,09
738 eC DATA 09,c1,3e,3a,78,06,b4,73,67,34
739 mS DATA 2d,04,06,0a,01,01,02,03,0a,47
740 zk DATA fa,ff,f4,e0,3f,d2,38,7a,04,db
741 p3 DATA 4b,b5,d0,b0,44,5c,cd,ff,f6,90
742 Vq DATA 44,18,33,50,05,48,84,72,ff,e9
743 HJ DATA 69,d8,45,53,44,82,40,e2,89,e2
744 pL DATA 17,51,ce,00,08,12,c7,42,47,7c
745 2F DATA 07,5e,f0,3f,82,f0,d6,44,1e,59
746 M6 DATA 13,ff,1d,ba,cc,06,1a,f0,02,16
747 11 DATA 41,50,37,c9,34,aa,22,08,24,f2
748 OQ DATA 7d,03,ee,e4,1a,e2,4a,80,67,58
749 hv DATA f2,c5,de,b6,9a,aa,c1,5a,b6,d9
750 kN DATA 7a,5a,30,70,14,99,3a,0d,d8,41
751 Ov DATA 84,56,ec,99,72,23,fa,e4,f6,31
752 GF DATA 29,5a,33,fa,e2,dc,da,be,9e,40
753 tg DATA ce,c9,2a,ba,18,43,51,01,b6,22
754 Jo DATA d5,58,de,d8,64,dc,42,80,b6,70
755 V2 DATA 80,7f,b2,43,c2,54,2e,1d,f4,b2
756 9k DATA 8a,23,19,50,66,04,26,8e,23,c7
757 1o DATA 74,21,53,7b,25,26,92,cb,ca,2a
758 1V DATA 06,63,03,22,c0,38,03,66,f4,8f
759 3K DATA 13,0e,20,c8,25,b4,88,04,c8,00
760 Qp DATA d0,85,b8,8e,ab,08,16,e9,66,28
761 QE DATA 28,5d,c8,eb,c0,35,c2,68,bb,1e
762 SD DATA 9a,a0,6e,20,4c,20,39,f9,ae,c2
763 Nn DATA 72,4e,60,60,4a,d7,6e,ea,66,2a
764 Bc DATA 60,bb,01,16,20,2d,b8,06,05,23
765 fh DATA c0,cc,73,da,f4,40,40,20,5d,58
766 Nm DATA 8d,20,f7,00,fe,c0,62,22,41,c0
767 gn DATA 30,60,1a,0c,40,3c,93,66,14,f8
768 Wv DATA da,20,1d,56,aa,47,c3,21,ba,0e
769 OM DATA 70,08,cb,4d,14,65,66,9d,b1,82
770 pD DATA 52,57,00,c8,70,04,4e,2d,03,b0
771 Ib DATA 75,2c,9c,cc,24,08,26,00,22,3a
772 xv DATA 00,c0,b4,fc,f5,d0,f4,5d,6e,a6
773 M6 DATA 33,6b,8a,80,28,b4,ce,2f,b3,6e
774 Vf DATA 67,7d,07,78,11,14,c0,89,a3,21
775 3V DATA c0,26,9b,16,03,eb,b9,2f,20,02
776 eS DATA bc,13,03,f3,bc,08,3a,0c,2f,98
777 Er DATA 38,7d,02,9e,a5,af,a3,14,1a,39
778 UU DATA 80,64,6f,e7,76,1c,80,64,66,31
779 JL DATA 3a,68,73,ef,03,68,26,09,e3,03
780 NP DATA 38,ec,08,03,f0,24,51,0c,09,e3
781 M6 DATA 00,d5,ca,58,8a,2b,4a,7a,0f,70
782 sG DATA 30,61,3b,bc,a2,3b,d6,14,40,bd
783 3M DATA 2c,78,31,03,8a,33,36,18,41,a7
784 v5 DATA 1b,2e,00,f6,05,ac,06,cd,03,2b
785 Da DATA 2b,0c,27,a1,85,51,fc,e5,59,f3
786 ct DATA 89,54,81,22,c1,e0,3f,60,f6,2f
787 Td DATA 0b,2c,07,53,86,26,f1,eb,4b,63
788 mo DATA 31,22,3c,c4,d1,3c,33,58,8b,14
789 gX DATA 53,c9,1e,67,04,08,c1,00,01,fa
790 u5 DATA 2a,72,3a,28,c0,58,8c,51,ce,ff
791 LJ DATA e0,26,5f,c7,fd,ed,22,e3,e9,a8
792 oX DATA c5,9a,f6,ea,a8,c5,90,f6,eb,a8
793 VT DATA c5,a8,f6,ec,a8,d3,b0,6e,b0,0c
794 Ji DATA 6b,d4,a3,bd,02,4e,71,2d,7a,00
795 DQ DATA 23,d6,be,86,6e,c6,2f,07,52,4f
796 bJ DATA ff,18,c3,11,10,5a,2b,73,52,87
797 De DATA 53,87,67,1a,e8,15,a5,04,a8,0e
798 C8 DATA 22,92,4b,28,00,08,58,80,e4,88
799 5G DATA 23,40,f9,11,48,60,e2,2e,1f,42
800 FI DATA a9,f4,a0,43,d2,57,2a,f6,50,50
801 uB DATA 68,06,e0,21,34,07,58,0b,41,32
802 iw DATA 4b,51,ac,84,11,34,ff,2e,e1,3e
803 WM DATA f9,2d,1d,61,28,e6,07,1b,e5,88
804 7U DATA 20,4b,d7,c0,fa,75,0c,22,64,89
805 80 DATA 22,71,94,ce,89,4e,ae,fd,90,20
806 Wz DATA 5f,3e,61,06,b4,81,85,40,ff,36
807 4w DATA 4a,85,67,02,52,86,7a,01,fa,51
808 Mn DATA 4a,9b,c4,52,20,06,e7,88,22,70
809 Ly DATA 08,f6,21,18,1b,67,24,b4,e7,89
810 Mb DATA 04,bf,0c,28,70,18,00,50,8c,53
811 2a DATA 80,20,4c,24,49,22,1b,10,80,ff
812 Vl DATA 46,22,12,d1,c1,24,88,51,c8,ff
813 Zs DATA f0,60,d8,e3,1f,06,f0,48,e7,ff
814 se DATA fe,4b,fa,01,80,41,fa,ff,f2,22
815 ea DATA 50,0b,c0,7f,d3,c9,58,89,2b,49
816 T9 DATA 80,1e,2b,7c,00,a3,cb,a5,18,61
817 kg DATA 0c,4e,b9,80,f7,81,4c,df,7f,ff
818 9z DATA d9,2d,20,26,14,2a,18,22,6d,00
819 Co DATA 18,d3,c5,d1,d0,91,01,ff,95,b1
820 g5 DATA 04,4a,60,6a,02,53,88,90,3f,14
821 3D DATA 51,5a,17,42,91,9d,21,38,76,03
822 IB DATA 6e,15,19,20,3c,82,78,ff,e5,6c
823 QL DATA 46,44,53,42,e5,ff,11,c0,5b,4a
824 6L DATA 43,67,06,b8,41,56,a1,ef,da,14
825 Vp DATA 21,96,04,48,82,d2,42,12,df,0a
826 Hr DATA 0a,43,2e,02,0e,07,04,01,13,20
827 L3 DATA 33,c8,00,df,f1,80,51,ff,57,f6
828 Ih DATA da,d6,26,6b,00,14,50,8b,b1,cb
829 Z5 DATA 6f,54,bf,c8,8e,29,3a,74,03,ec
830 ir DATA 10,01,ca,bc,1a,42,41,52,42,5e
831 5v DATA 00,67,12,b5,29,53,43,f8,c3,51
832 oM DATA 2e,03,16,33,20,05,48,83,d2,43
833 wQ DATA c6,3f,0c,0a,02,c2,fe,d4,0a,06
834 P6 DATA 04,03,02,0c,41,f5,69,67,42,47
835 wt DATA d6,67,34,76,01,6a,0c,00,3c,16
836 Mb DATA 52,43,cc,2a,18,33,dc,48,84,28
837 ZL DATA 6e,10,c0,cc,6b,e3,4b,d4,73,30
838 gk DATA 04,d3,0b,00,32,0b,04,07,88,4f
839 P5 DATA d5,00,9e,42,42,76,05,42,44,b2
840 PV DATA 37,ea,64,04,76,08,78,40,e3,08
841 9F DATA 66,04,10,20,e3,10,e3,52,51,cb
842 47 DATA ff,f4,d4,44,45,f1,09,fc,ff,48
843 KF DATA c1,d5,c1,53,41,13,22,51,c9,ff
844 sq DATA fc,60,52,7f,c8,4e,75,95,76,b3
845 ev DATA 2d,a0,5a,a8,f8,06,00,fc,00,fe
846 ut DATA 00,82,e8,87,8c,00,f4,00,86,00
847 dh DATA fa,00,93,00,fd,00,69,80,6e,80
848 8w DATA 04,c0,07,40,02,60,03,a0,01,40
849 Ov DATA 01,c0,dd,c0,7f,00,00,e7,28,24
850 r0 DATA a8,41,26,0d,28,6a,2a,50,83,2c
851 Mx DATA 1a,2e,d4,a0,3e,66,01,32,39,34
852 PQ DATA a8,41,b5,ea,01,8a,41,40,ba,01
853 KA DATA 3c,2d,07,15,a5,01,ab,cf,88,18
854 70 DATA 11,92,00,38,00,94,00,d0,00,8e
855 X2 DATA 29,81,00,90,29,c1,00,e0,21,c0
856 8w DATA ff,a0,08,75,e4,b6,3b,56,84,2e
857 Ye DATA 62,01,02,cc,45,04,00,24,01,3d
858 q3 DATA b9,16,da,5c,c7,82,05,79,01,84
859 YR DATA 09,bd,01,86,02,46,01,a2,0f,ca
860 tl DATA 01,a4,0d,22,01,a6,1a,f0,df,29
861 Ex DATA c0,6a,31,b9,c5,39,72,6b,e5,41
862 10 DATA 96,ca,49,ad,94,51,1b,28,59,37
863 Iv DATA 4e,61,6e,98,69,dc,24,71,b9,39
864 dY DATA 79,72,53,e4,81,66,c8,89,4d,90
865 YB DATA 91,9b,cd,99,4d,e6,a1,0a,0f,b1
866 fa DATA 71,16,e2,0e,b9,2c,c4,0d,c1,59
867 Fk DATA 88,0c,c9,b3,10,0b,d1,67,21,0a
868 3Y DATA d9,ce,42,09,e1,9c,95,08,e4,38
869 MM DATA 13,00,22,00,e9,d9,85,07,f1,38
870 27 DATA 0b,06,f9,71,36,e2,05,ff,df,1d
871 h9 DATA 94,10,04,09,67,21,03,11,ce,42
872 Xh DATA 02,19,9c,85,01,21,09,78,17,80
873 sG DATA 55,71,ff,ff,ff,fe,01,0f,bd,6b
874 xt DATA cc,c1,07,8b,18,6f,f0,1e,12,30
875 kE DATA b5,23,09,44,46,31,3a,ff,21,0b
876 o4 DATA 4f,40,7f,bc,1f,0d,43,41,3c,07
877 HH DATA 4c,56,97,09,19,8c,e0,0a,0d,46
878 y9 DATA f0,0b,06,23,0c,78,87,0f,3c,01
879 zC DATA 0d,29,0c,43,f0,0b,2e,57,77,c5
880 MP DATA 4b,a6,83,24,ff,0e,0c,45,58,90
881 7e DATA 17,1f,c0,98,43,4c,49,3b,13,c6
882 D9 DATA 9d,01,20,b0,57,1a,3e,3e,8c,43
883 Yu DATA 7b,62,80,39,c6,79,20,56,33,2e
884 Ws DATA b8,a3,3c,3c,3d,3d,2d,2d,2d,20
885 vX DATA 2c,fe,03,13,59,c8,8a,82,3c,50
886 LX DATA fa,42,b4,30,16,52,41,48,4d,eb
887 Ok DATA 08,4d,49,e4,41,45,4c,94,66,f0
888 8H DATA 20,f4,52,32,49,05,fe,c0,43,4f
889 g0 DATA 50,59,52,49,47,48,8d,ff,42,c0
890 7X DATA 71,4d,39,4b,22,45,26,e4,f9,45
891 rH DATA 43,48,4e,49,4b,7a,08,4c,4f,8b
892 1V DATA a6,10,23,19,08,ce,44,31,33,e2
893 Lb DATA 05,2a,38,8c,ff,18,0a,50,75,17
894 Tb DATA 46,70,6f,0e,84,53,ff,83,23,e7
895 bi DATA 05,11,de,aa,00,ef,85,02,12,6f
896 Tj DATA 55,80,f7,42,81,13,b7,0d,6e,50
897 cA DATA 73,b1,c0,14,5b,15,e0,55,40,1b
898 Wo DATA 0f,99,12,f0,15,56,05,78,20,f6
899 Q4 DATA 87,44,b8,44,e5,b5,53,53,58,9b
900 As DATA 16,2b,53,3e,05,24,54,49,4f,4e
901 2e DATA 20,88,ff,21,54,48,25,06,b2,35
902 zS DATA 4a,12,af,ff,01,18,9d,0b,e3,20
903 NZ DATA 44,55,4d,50,bc,9a,d8,c0,62,52
904 g1 DATA 43,17,5f,50,54,f0,a0,20,b4,24
905 mN DATA b0,cb,00,ee,09,72,41,b8,6c,3a
906 Eq DATA 53,59,c1,e3,91,c0,3b,c1,c0,6b
907 1T DATA 90,83,b6,c0,24,bb,60,1f,0e,67
908 T2 DATA 2f,78,67,40,1e,58,20,c1,42,08
909 zp DATA 54,22,20,02,9e,05,09,49,53,20
910 T8 DATA 41,4c,9e,7f,41,44,59,52,7f,5b
911 xv DATA 13,1d,77,80,c7,02,00,0e,90,02
912 Nd DATA b1,ed,00,cf,c9,00,4f,c0,46,2e
913 PO DATA 96,00,4a,1d,02,5e,01,d6,52,4e
914 Az DATA c3,06,72,3d,20,4e,4f,b8,65,53
915 ea DATA 41,c4,68,44,92,89,00,af,00,d3
916 2V DATA 43,52,61,43,48,16,fe,47,91,4b
917 sp DATA 50,96,41,53,84,43,57,41,49,54
918 mv DATA 7c,44,00,6f,44,01,41,42,cf,0d
919 lu DATA 20,54,4f,20,52,45,50,41,43,f9
920 aZ DATA 1d,64,40,4c,45,ac,cd,71,07,4f
921 Uk DATA 55,54,f8,96,46,20,4d,45,4d,e1
922 o1 DATA 2f,59,a4,32,60,03,20,20,ca,27
923 1V DATA 01,0e,20,bd,52,e1,2e,2e,2e,20
924 bH DATA 84,27,44,45,46,49,4e,f0,4f,48
925 vL DATA 55,4e,4b,3d,52,20,4f,56,45,52
926 Jy DATA 4c,41,59,45,44,20,00,1f,14,00
927 Rz DATA 80,ff,43,ff,1f,18,bd,81,0c,c0
928 Mm DATA 2e,67,4d,70,68,69,63,73,2e,6c
929 xo DATA 69,62,72,61,72,79,e8,7f,1a,37
930 ob DATA 03,0c,39,1b,30,43,f8,1e,72,f0
931 Ou DATA 1c,31,38,3d,e4,e0,1d,32,34,bb
932 EX DATA a8,c1,ff,01,1e,33,3b,b5,04,0f
933 9N DATA 78,c0,1d,50,40,20,30,30,00,74
934 qb DATA 6f,70,61,7a,2e,66,6f,6e,74,fd
935 uC DATA 13,31,d6,00,b9,93,66,c5,7a,e6
936 Ws DATA 07,44,46,30,3a,79,20,32,0f,80
937 Kc DATA 2a,ef,58,05,0f,af,96,40,9e,f3
938 WI DATA 83,63,6f,6e,73,6f,6c,65,2e,64
939 8X DATA 65,76,69,63,65,f7,1f,c4,e1,19
940 1u DATA c0,02,ea,f0,80,18,3c,e0,01,0f
941 XV DATA b8,8b,e2,94,88,00,49,4e,50,55
942 9p DATA f6,69,ec,34,1f,a8,5a,15,92,70
943 Jg DATA 0e,9e,6a,da,50,05,15,0c,16,50
944 d0 DATA 20,07,c6,35,ce,a8,02,22,d6,1c
945 3x DATA 06,d4,a0,3e,0a,23,44,70,d0,50
946 OF DATA 83,e6,1a,f0,d4,a0,f6,0a,24,fe
947 ax DATA 70,08,50,83,0e,1a,16,d4,a0,1e
948 xq DATA 06,30,35,48,a8,41,52,0d,58,6a
949 Py DATA 5e,50,83,9a,1a,ac,54,25,cc,01
950 cZ DATA 0e,36,aa,26,62,00,0b,31,e2,30
951 OS DATA 50,46,29,28,eb,aa,00,01,00,00
952 yo DATA 03,f2,40,f8,00,00,00,50,00,00
953 Em DATA 03,eb,00,00,0d,b1,00,00,03,f2
954 da DATA 00,00,03,e8,00,00,00,03,50,49
955 xe DATA 55,53,00,00,24,7c,c7,a3,fb,0c
956 ox DATA 00,00,03,f2
(C) 1990 M&T
```

Listing. »MasterCruncher_Gen« (Schluß)

Es existiert bereits eine Vielzahl von Anti-Virus-Programmen, doch jedes hat seine Schwächen. Eines findet nur bestimmte Viren, ein anderes hilft nicht gegen Link-Viren (dazu später). Mit »VirusControl« bieten wir Ihnen ein optimales Werkzeug gegen Viren jeder Art an. VirusControl ist trotz seiner Leistungsvielfalt einfach zu bedienen, denn gerade für Einsteiger sind Viren auf dem Amiga gefährlich. Erstens weiß er die Anzeichen nicht zu deuten, und zweitens ist er meist nicht in der Lage, sie zu beseitigen.

Damit Sie die einzelnen Erklärungen richtig verfolgen können, sollten Sie zuerst das Programm besitzen. Folgende Schritte sind erforderlich, um VirusControl zu installieren:

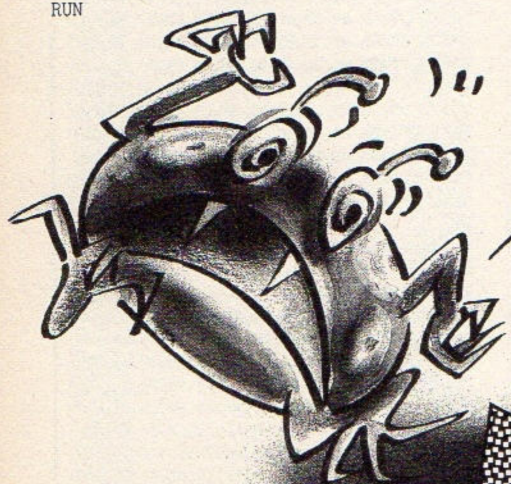
— Stellen Sie sicher, daß auf der Zieldiskette mindestens 35 Blocks frei sind. Die Anzahl der freien Blocks erfahren Sie, wenn Sie in einem CLI-Fenster den Befehl INFO eingeben. Wenn nötig, müssen Sie unbenötigte Dateien von der Diskette löschen.

— Starten Sie den Amiga-Basic-Interpreter durch Doppelklick. Geben Sie folgende Kommandos ein:

```
CLEAR ,50000
LOAD "VirusControl_Gen"
```

— Starten Sie das Programm mit dem Kommando:

```
RUN
```



DER ENTSEUCHUN

de Boot entdeckt

— Wenn das Programm den Text »Bitte Workbench-Diskette in DF0: einlegen!« ausgibt, legen Sie Ihre Workbench-Diskette ein, mit der Sie normalerweise den Amiga starten.

Auf dieser Diskette befindet sich nun im Verzeichnis C das fertige Programm. Außerdem ist die »Startup-Sequence« verändert worden, so daß VirusControl automatisch gestartet wird.

Um sicherzugehen, daß bei Ihnen nicht schon ein Virus im System ist, schalten Sie Ihren Amiga nach Beendigung des Programms für 20 Sekunden aus. Booten Sie anschließend mit der behandelten Workbench-Diskette. Sollte nun etwas Außergewöhnliches passieren, lesen Sie weiter, denn es folgt die Erläuterung der Funktionen von VirusControl.

VirusControl wird im CLI oder in der »Startup-Sequence« gestartet. Auch ein Start von der Workbench ist möglich. Die Parameter werden entweder beim Start im CLI angegeben oder in die ».info«-Datei im Feld »TOOL TYPES« eingetragen.

Beim Versuch, VirusControl zu starten, kontrolliert das Programm, ob der Amiga bereits infiziert ist. Hierzu werden einige System-Vektoren überprüft. Wenn diese verändert sind, wird VirusControl nicht installiert. Statt dessen erscheint die Aufforderung, den Amiga auszuschalten, wieder einzuschalten und möglichst frühzeitig VirusControl zu starten. Nun kann kein Virus mehr unerkannt in das System eindringen. Bei Veränderungen an einigen bestimmten System-Vektoren gibt VirusControl nur eine Warnmeldung aus, da diese auch von »seriösen« Programmen (Facc II, XOper etc.) verbogen werden. Ferner prüft VirusControl, ob es bereits installiert wurde, und gibt eine entsprechende Meldung aus.

Bei einem Reset wird geprüft, ob die linke Maustaste gedrückt ist. Ist dies der Fall, wird VirusControl entfernt. Ansonsten zeigt ein weiß-blaues Farbsignal an, daß VirusControl weiterhin das System kontrolliert. Ab jetzt prüft das Programm, ob ein Zugriff auf eine Diskette ab Track 0 (erster Teil der Bootsektoren) erfolgt. Dies ist beim gleich folgenden Booten und auch beim normalen Einlegen einer Diskette der Fall. Auch der Befehl LOADWB greift auf die Bootsektoren zu. Wird VirusControl ohne »-t« aufgerufen, erfolgt auch hier eine Überprüfung durch VirusControl.

Der Amiga bootet nur dann von einer Diskette, wenn folgende zwei Bedingungen erfüllt sind: Die Diskette muß die DOS-Kennung besitzen und die Bootblock-Checksumme muß stimmen. Ist der Bootblock der Diskette kein Standard-Bootblock, handelt es sich um eine »verdächtige Disk«, da es ein Bootblock-Virus sein kann. Es kann sich aber auch um ein harmloses Boot-Intro (z. B. eine Grafik von Bootgirl, AMIGA-Magazin, Ausgabe 3/88) oder ein selbststartendes Spiel handeln.

Wenn der Amiga von einer Diskette booten will, prüft VirusControl, ob die eingelegte Boot-Diskette verdächtig ist. Wenn ja, erscheint ein Dialogfeld (Requester), das eine gefahrlose Bootsimulation anbietet, oder aber die echte Disk-Boot-Routine. Im ersten Fall wird das Programm in den Bootsektoren nicht durchgeführt. Wird die zweite Möglichkeit gewählt, erscheint ein weiteres Dialogfeld, das auf die Gefahr des Aktivierens von Bootblock-Viren durch das »echte« Booten hinweist. Es gibt Spiele, bei denen der Bootblock unbedingt abgearbeitet werden muß, damit das Spiel läuft. In diesem Fall ist es nötig, das Programm in den Bootblocks laufen zu lassen. Wählen Sie diese Möglichkeit jedoch nur, wenn Sie sicher sind, daß es sich nicht um einen Virus handelt!

Wird während der Arbeit mit dem Amiga eine verdächtige Diskette eingelegt, wird dieses durch das Erscheinen eines Kontrollfensters (siehe Bild 1) deutlich gemacht. Hält man während des Disk-Einlegens die rechte Alt-Taste gedrückt, wird die Diskette nicht durch VirusControl überprüft. Drückt man während des Disk-Einlegens die linke Alt-Taste, erscheint auch bei nicht verdächtigen Disketten das Kontrollfenster.

Falls Sie nur über ein Laufwerk verfügen, ist es notwendig, das Kontrollfenster auf andere Art zu öffnen. Rufen Sie VirusControl im CLI oder von der Workbench auf und drücken Sie sofort danach die linke Alt-Taste. Dadurch ist die Diskette im Laufwerk nicht blockiert und Sie können mit VirusControl diese Diskette behandeln.

Kann aufgrund von Speichermangel das Kontrollfenster nicht geöffnet werden, zeigt VirusControl dies durch eine kurze Schwarzfärbung des Bildschirms an. Das Kontrollfenster bietet eine Vielzahl nützlicher Fähigkeiten an, die durch Anklicken des entsprechenden Schalters (Gadgets) ausgewählt werden.

Das Anklicken des Schließschalters (Close Gadget, links oben im Kontrollfenster) beendet die Arbeit mit der eingelegten Disk. Erst danach darf eine andere Diskette eingelegt werden. Solange der Schließschalter nicht betätigt wurde, darf die Diskette lediglich zum Ändern des Schreibschutzes entnommen werden. Mit den beiden Schaltern rechts oben, können Sie das Kontrollfenster vor oder hinter die anderen Fenster auf der Workbench legen.

Beende VirusControl

Das Anklicken von »Beende VirusControl« entfernt VirusControl aus dem Speicher. Danach kann die Diskette entnommen werden.

Kill System-Virus

Dieser Programmteil versucht, einen aktiven Virus zu eliminieren, indem unter anderem verschiedene Systemvektoren restau-



Hatten Sie auch schon Ärger mit Viren? Mit »VirusControl« ist die Zeit Sorgen vorbei. Sowohl block- als auch Link-Viren das Programm.

GSTRUPP

riert werden. Es ist allerdings nicht garantiert, daß der Amiga nach dem Versuch richtig weiterarbeitet.

Bootblock -> Schirm

Ein Anklicken von »Bootblock -> Schirm« zeigt den gesamten Bootblock (1024 Byte) als ASCII-Text an. An verräterischen Texten können Sie die meisten Viren erkennen.

Bootblock -> File

Schreibt den gesamten Bootblock(1024 Byte) in eine Datei, deren Name im Texteingabefeld (Stringgadget, ganz unten im Kontrollfenster) eingegeben wird.

File -> Bootblock

Ein Anklicken von »File -> Bootblock« überschreibt den Bootblock mit einer Datei, deren Name im Texteingabefeld eingetragen ist. Mit dieser Möglichkeit, den Bootblock in ein File zu speichern, und dieses File auch wieder auf den Bootblock zu schreiben, können Sie von allen wichtigen Disketten eine Kopie des Bootblocks anfertigen, und somit notfalls den Original-Bootblock wiederherstellen und wieder mit der Diskette arbeiten.

Standard-Bootblock

Haben Sie einen Virus entdeckt, wollen Sie ihn auch beseitigen. Durch das Anklicken von »Standard-Bootblock« wird der Original-Amiga-Bootblock auf Diskette geschrieben. Das entspricht dem Befehl INSTALL.

NoFastMem-Bootblock

Durch Anwählen dieses Schalters schreibt VirusControl eine Bootblock-Routine auf Diskette, die das Fast-Memory schon zu Beginn des Boot-Vorgangs abschaltet. Der Rechner verhält sich so, als ob er nur Chip-Memory besitzen würde. Ein blaues Farbsignal zeigt die NoFastMem-Routine an. Hält man während des Resets die linke Maustaste gedrückt, wird die NoFastMem-Routine nicht ausgeführt.

f.Checksum -> NoBoot

Schreibt eine falsche Bootblock-Checksumme auf die Diskette, wodurch von der Diskette nicht mehr gebootet werden kann. Ein eventueller Bootblock-Virus ist dadurch (vorübergehend) unschädlich gemacht.

r.Checksum -> NoBoot

Ist das Gegenstück zu »f.Checksum -> NoBoot«, es korrigiert die Checksumme wieder. Dadurch kann mit der Diskette wieder gebootet werden, und somit auch ein eventueller Virus aktiviert werden. Die beiden Routinen ändern nur die Bootblock-Checksumme. Der Bootblock-Inhalt bleibt erhalten.

in S:NoWarning-File

Wie schon erwähnt, gibt es ungefährliche Bootblöcke, die VirusControl als »verdächtig« einstuft. VirusControl wäre unkomfortabel, wenn in diesem Fall jedesmal das Kontrollfenster erscheinen würde. Mit diesem Schalter können Sie diesen Vorgang verhindern. Wenn Sie eine Diskette einlegen, von der Sie GARANTIERT wissen, daß sich kein Virus darauf befindet, sondern ein ungefährlicher Bootblock, klicken Sie diesen Schalter an. Der verdächtige Bootblock wird nun in der Datei »s:NoWarning« gespeichert. Existiert s:NoWarning, hängt VirusControl den Bootblock an das Dateiende an. Andernfalls wird S:NoWarning zuerst erstellt. Sollte der

Wichtige Hinweise

Beim Starten folgender Programme öffnet VirusControl ein Warn-Gadget. Sie können es ignorieren, da diese Programme die folgenden Vektoren legal nutzen:

Programm:	Vektoren:
Setpatch	Alert AllocEntry UserState ab V 34,27
Xoper	Switch
CPr	AddTask RemTask OpenDevice
NoFastMem	AllocMem AvailMem
ALF Autoboot- Controller	KickMemPtr KickTagPtr KickCheckSum

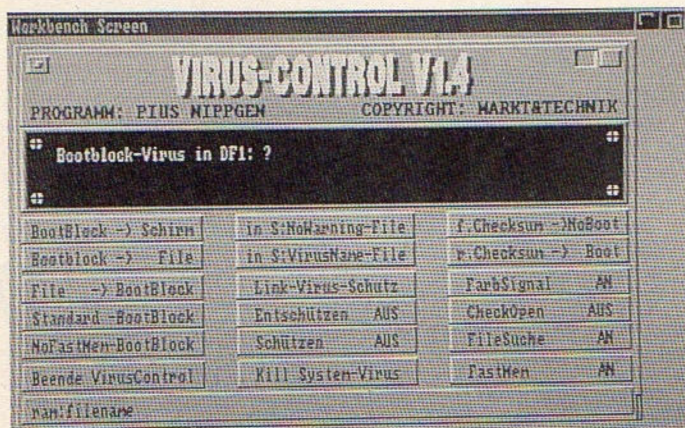


Bild 1. Das Kontrollfenster von »VirusControl« erscheint bei »verdächtigen« Disketten

zu speichernde Bootblock bereits in s:NoWarning vorhanden sein, erscheint ein Requester, der die Möglichkeit anbietet, den Bootblock aus der Datei zu entfernen. Legen Sie nun später eine verdächtige Diskette ein, deren Bootblock in s:NoWarning vorhanden ist, blitzt der Bildschirm nur kurz rot auf. Das Kontrollfenster wird nicht geöffnet. Noch eine Warnung: Übernehmen Sie diese Datei NIE von jemand anderem. Vielleicht will er Sie ärgern!

in S:VirusName-File

Haben Sie einen Virus identifiziert, können Sie in die Datei »s:VirusName« eintragen lassen. Es wird zusätzlich ein Name für den Bootblock gespeichert. Dieser Name wird wieder mittels des String-Gadgets eingegeben. Existiert s:VirusName, wird der Name und der Bootblock an das Dateiende angehängt, andernfalls wird s:VirusName zuerst erstellt. Sollte der abzuspeichernde Bootblock bereits in s:VirusName vorhanden sein, erscheinen Requester, die folgende Möglichkeiten anbieten: Man kann den Bootblock und Namen aus s:VirusName entfernen. Man kann den alten Namen durch den neuen Namen ersetzen. Legen Sie nun eine Diskette ein, deren Bootblock in S:VirusName gespeichert ist, wird im Kontrollfenster der Virusname ausgegeben. Mit dieser Methode sind Sie in der Lage, VirusControl ständig auf den neuesten Stand zu bringen.

Wenn eine verdächtige Disk eingelegt wird, durchsucht VirusControl S:NoWarning und danach S:VirusName. Legen Sie eine Diskette bei gedrückter linker Alt-Taste ein, wird nur s:VirusName durchsucht, da dann immer das Kontrollfenster erscheint. Man kann »s:« mittels ASSIGN auf ein schnelleres Medium umlenken, da das Durchsuchen von 100 KByte (100 Bootblöcke) auf Diskette ca. 10 Sekunden dauert. In RAM:, VD0: oder DH0: benötigt VirusControl dafür nur noch ca. 2 Sekunden. Wenn Sie s: auf die RAM-Disk legen, vergessen Sie nicht, die Dateien am Ende wieder dauerhaft zu speichern. Das Durchsuchen der Files unterbleibt, wenn Sie VirusControl mit dem Parameter »-f« starten. Sind die Dateien

Den Dateizugriff im Griff

in s: nicht vorhanden, unterbleibt zwangsläufig ein Durchsuchen. Alle tiefergehenden Funktionen wie Bootblock beschreiben, VirusControl entfernen oder System-Virus entfernen, sind zusätzlich durch weitere Requester gegen unabsichtliche Ausführung gesichert. Weiterhin zeigen Requester schreibgeschützte Disketten an. Ein Fehler beim Öffnen einer Datei wird ebenfalls durch Requester angezeigt. Da bis zu vier Laufwerke möglich sind, kann VirusControl bis zu vier Kontrollfenster gleichzeitig bearbeiten. Diese sind daher mit dem dazugehörigen Laufwerk gekennzeichnet.

Ein weiß-blaues Farbsignal signalisiert bei jeder neu eingelegten Diskette (oder beim LOADWB-Befehl oder beim erneuten Starten von VirusControl) das korrekte Arbeiten von VirusControl. Dieses optische Zeichen kann jedoch abgeschaltet werden, indem man VirusControl zusätzlich mit »-c« aufruft. Um das Kontrollfenster besser zur Geltung zu bringen, werden vorübergehend neue Workbench-Farben gesetzt und am oberen und unteren Bildschirmrand ein blauer Rahmen verwendet.

Beim Starten von VirusControl überprüft das Programm die eingelegten Disketten. Bei Angabe von »-t« wird dieses unterlassen.

Geben Sie den Parameter »-o« ein, bewirkt dies, daß vor jedem Filezugriff ein Requester erscheint. Mit ihm können Sie den Zugriff erlauben oder verbieten. Hält man jedoch die Ctrl-Taste gedrückt, wird direkt zur Original-Routine gesprungen. Diese Dateizugriffskontrolle kann sehr nützlich für das Erkennen von Link-Viren sein. Link-Viren breiten sich nicht über den Bootblock aus, sondern sie hängen sich an bestehende Dateien. Sie verlängern also diese Dateien und werden durch das Aufrufen derselben aktiviert. Wenn sich nun ein Link-Virus an ein Programm hängen will, kann der Filezugriff mittels des Requesters erkannt und verhindert werden. Diese Option ist nur zum Suchen gedacht, wenn Sie vermuten, daß sich ein Link-Virus eingeschlichen hat.

Link-Virus-Schutz

VirusControl bietet einen weiteren äußerst wirksamen Schutz vor Link-Viren. Das Datei-Schutz-System wird mit dem Gadget »Link-Virus-Schutz« aufgerufen. Es wird die im String-Gadget angegebene Datei nach der VirusControl-Schutz-Kennung durchsucht. Weist das File noch keine Schutz-Kennung auf, schützt Virus Control (sofern es sich um ein Programm handelt) die Datei automatisch. Die Schutzfunktion läßt sich auch auf ganze Unterverzeichnisse anwenden. Wenn man VirusControl mit »-s« aufruft, oder wenn man »Schützen An« durch Anklicken setzt, wird eine Datei erst nach Rückfrage geschützt. Dateien, die nur Daten enthalten, werden generell nur nach Rückfrage geschützt, da es nicht sinnvoll ist, z. B. eine Textdatei zu schützen. Anstatt eines einzelnen Dateinamens sind auch Verzeichnis-Angaben erlaubt. Man kann also komfortabel das komplette C-Verzeichnis schützen lassen. Das automatische Durcharbeiten eines kompletten Verzeichnisses kann jedoch jederzeit durch das gleichzeitige Drücken von linker und rechter Shift-Taste gestoppt werden.

Weist eine Datei bereits die Schutzkennung auf, wird nun geprüft, ob die Datei verändert wurde. Ist dies der Fall, erscheint ein Requester, der auf den vermutlichen Link-Virus-Befall hinweist. Mittels des Requesters ist es nun möglich, den Schutz der Datei wieder zu entfernen. Beim nächsten Durchgang wird diese Datei mit einer neuen Kennung versehen. Mit File-Entschützen kann aber ein Link-Virus meist nicht entfernt werden. Leider sind wegen des flexiblen Aufbaus für Programmdateien viele Link-Viren-Typen denkbar. Es ist also unmöglich, ein generelles Programm zum Entfernen von Link-Viren zu schreiben. Vielmehr muß man für jeden Link-Virus gezielt ein spezielles Programm erstellen.

Es empfiehlt sich also folgende Vorgehensweise, um sich vor Link-Viren zu schützen:

— Sicherheitskopien der Originaldateien anfertigen
— Originaldateien schützen (mit »Link-Virus-Schutz«)
— Regelmäßig prüfen, ob die Dateien noch unverändert sind.
— Wird eine Veränderung festgestellt, deutet dies auf einen Link-Virus hin. Man entfernt nun diesen Link-Virus einfach dadurch, indem man das verdächtige Programm mit dem Original-Programm überschreibt. Dieses neue Programm sollte nun sofort wieder geschützt werden.

Durch das Anklicken von »Link-Virus-Schutz« veranlassen Sie VirusControl, noch ungeschützte Dateien zu schützen und bereits geschützte auf Veränderung zu überprüfen. Dies geschieht weitgehend automatisch. Wird jedoch »Schützen An« gesetzt, muß das »File schützen« extra bestätigt werden. Es empfiehlt sich, beim erstenmal die Verzeichnisse automatisch schützen zu lassen. Danach sollte man aber »Schützen An« setzen, denn dadurch wird man in Zukunft extra mittels Requester darauf aufmerksam gemacht, wenn eine neue Datei im Verzeichnis vorliegt. Ein Virus könnte durchaus ein gefährliches Programm erstellen, bzw. ein vorliegendes Programm einfach überschreiben. Dieses kann nun nicht mehr übersehen werden. Wenn eine Veränderung festgestellt wird, muß auch diese wegen eines möglichen Link-Virus-Befalls extra bestätigt werden. Wenn Sie VirusControl mit »-e« aufrufen oder »Entschützen An« durch Anklicken setzen, bietet VirusControl die Möglichkeit, die Datei zu entschützen, wodurch das Original-Programm wieder zur Verfügung steht. VirusControl zeigt alle File-Schutz-Vorgänge am Bildschirm an. Ferner wird ein dauerhaftes Protokoll in der Datei »s:VCprotokoll« angelegt.

Man sollte die Verzeichnisse »C:«, »L:«, »LIBS:«, und »DEVS:« schützen, da hier ausführbare Programme enthalten sind, die entweder von Benutzer selbst oder vom Betriebssystem aufgerufen werden. Bei diesem Aufruf könnte dann ein Link-Virus aktiviert werden. Info-Dateien werden generell nicht geschützt.

Die Kurzanleitung zu VirusControl kann durch erneutes Starten

von VirusControl abgerufen werden. Hierbei können auch die Parameter (»-c«, »-f«, »-t«, »-o«, »-m«, »-e« und »-s«) neu gesetzt werden. Geben Sie keine Parameter an, verwendet VirusControl die Standard-Einstellung:

— Farbe AN
— FileSuche AN
— Kein Test bei Datei öffnen
— ChipMemFirst AUS
— Entschützen AUS
— Schützen AUS

Die Parameter können auch mittels der entsprechenden Gadgets (FarbSignal, CheckOpen, FileSuche, ChipMemFirst, Entschützen, Schützen) neu gesetzt werden.

Kontrolle der Systemvektoren

VirusControl kann also mit Sicherheit das Eindringen von Bootblock-Viren verhindern, da es verdächtige Disketten anzeigt und eine gefahrlose Bootsimulation anbietet. Ferner zeigt VirusControl mittels des Dateischutz-Systems Link-Viren an.

Außer allen bisher genannten Maßnahmen installiert Virus Control einen separaten Task zur Überprüfung der System-Vektoren. Wird eine Veränderung festgestellt, zeigt VirusControl mittels Requester die jeweilige Veränderung an. Nach Anklicken von »ENDE« wird normal weitergearbeitet. Ein Anklicken von »VIRUS-ENTFERN-VERSUCH« bewirkt, daß einige Vektoren auf den Standardwert gesetzt werden. Diese Option kann jedoch zu einem Absturz führen, da VirusControl mit dem Virus kollidiert. Es ist daher ratsam, den Virus durch Ausschalten des Amiga zu entfernen, denn nur diese Methode kann wirklich sicher sein. Sollte man es einem Virus erlaubt haben, sich im System zu installieren, indem man im »KILL SYSTEM VIRUS«-Requester »ENDE« anklickt, kann man dennoch später jederzeit versuchen, den Virus zu entfernen, bzw. VirusControl wieder voll zu installieren. Dieses ist durch den separat laufenden VirusControl-Task möglich. Durch gleichzeitiges Drücken von linker und rechter Alt-Taste wird der »KILL SYSTEM VIRUS«-Requester aufgerufen.

Änderungen an den Vektoren können aber auch von seriösen Programmen herrühren. So »verbiegt« das Disketten-Cache-Programm FACC ebenfalls den BeginIO-Vektor der trackdisk.device-Struktur. Die resetfeste RAM-Disk RAD: benutzt die KICK-Vektoren. Der Setpatch-Befehl verbiegt den Alert-Vektor und AllocEntry-Vektor.

Um herauszufinden, ob es sich bei einem verdächtigen Bootblock wirklich um einen Virus handelt, kann man folgendermaßen vorgehen: Man bootet mit installiertem VirusControl von der verdächtigen Disk, aktiviert also bewußt einen eventuellen Virus. Alle eingelegten Disketten sollten schreibgeschützt sein, damit sich der eventuelle Virus nicht verbreiten kann. Handelt es sich um einen Virus, kann man dies an folgenden Umständen erkennen:

— Der Vektor-Kontroll-Task weist mittels Requester auf veränderte Vektoren hin.
— Ein Schreibzugriff auf den Bootblock wird durch einen Requester angezeigt.
— Kein weiß-blaues Farb-Signal beim Einlegen einer neuen Disk, obwohl das Farb-Signal eingestellt ist.
— Beim Disk-Einlegen erscheint trotz gedrückter linker Alt-Taste kein Kontrollfenster.
— Bei einem Reset tritt kein weiß-blaues Farb-Signal auf.
— Beim Aufruf des VirusControl-Programmes wird eine Meldung ausgegeben, die auf die vermutliche Virus-Infektion hinweist.

Da sich der Amiga flexibel programmieren läßt, gibt es auch für ein Virus-Programm viele Möglichkeiten. Es werden daher nicht alle eben genannten Effekte auftreten. Es kann auch zu Kollisionen zwischen VirusControl und dem Virus kommen, die zum Absturz führen. Anschließend muß der Amiga ausgeschaltet werden, da dies die einzig sichere Methode ist, um einen aktiven Virus zu entfernen.

Die einzige sichere Methode, um einen Bootblock-Virus zu identifizieren, ist zugleich auch die gefährlichste. Man bootet ohne VirusControl von der verdächtigen Disk, und legt eine nicht schreibgeschützte, nicht infizierte Diskette ein. Nun schaltet man den Amiga aus, wartet ca. 20 Sekunden, schaltet den Amiga wieder ein, bootet von einer sauberen Disk, startet VirusControl und legt die nicht schreibgeschützte Diskette ein. Wenn die Diskette nun verdächtig geworden ist, handelt es sich um einen Virus. Die Diskette sollte sofort »überschrieben« werden, damit man später nicht versehentlich von dieser Diskette bootet, und somit den Virus aktiviert und dadurch weiterverbreitet.

Da VirusControl nur wenig Rechenzeit in Anspruch nimmt, ist es empfehlenswert, das Programm immer laufen zu lassen. Es beeinträchtigt Ihre Arbeit am Computer nicht, bringt Ihnen aber eine hohe Sicherheit gegen Viren jeglicher Art.

Die Gefahr der »Infektion« ist auf PC- und Macintosh-Systemen höher als auf dem Amiga. Damit dies so bleibt und die Gefahr noch geringer wird, sollten Sie VirusControl unbedingt einsetzen. Denn auf keinem anderen Computer gibt es ein so leistungsfähiges Programm zum Aufspüren dieser lästigen Viren. *P. Nippgen/R. Beaupoil/kn*

Viren bringen Computer in den Ruf der Unzuverlässigkeit. Das kann nicht im Interesse der Amiga-Benutzer sein. Es hat sich schon gezeigt, daß Viren-Programmierer auch in der Lage sind, »vernünftige« Programme zu entwickeln. Es wäre schade, wenn folgender Ausspruch von Jim Mackraz (Amiga Software-Entwickler, arbeitet zur Zeit am Intuition-Teil der Kickstart 1.4) noch lange Gültigkeit besitzt: "Die beste Software aus Deutschland kommt meistens im Bootblock!"

Programmname:	VirusControl_Gen
Computer:	A500, A1000, A2000 mit Kickstart 1.2 und 1.3
Sprache:	Amiga-Basic 1.2
Bemerkung:	Erzeugt lauffähiges Programm

```
Programmautor: Pius Nippgen
---------------------------
OmO REM Generiert lauffähiges Progr
    amm
LI  ON ERROR GOTO Fehler
bh  CLS
VT  PRINT "Bitte Workbench-Diskette
     in DF0: einlegen!"
b5  a$=""
Wm  WHILE a$="":a$=INKEY$:WEND
9v  OPEN "DF0:c/VirusControl" FOR O
    UTPUT AS 1
FW  READ anz
se  FOR i=1 TO anz
7r1  READ h$
2F2   wert1=ASC(LEFT$(h$,1))
fT    IF wert1>64 THEN wert1=wert1-
      87 ELSE wert1=wert1-48
JM    wert1=wert1*16
Bg    wert2=ASC(RIGHT$(h$,1))
Ot    IF wert2>64 THEN wert2=wert2-
      87 ELSE wert2=wert2-48
Tm    wert=wert1+wert2
DK    PRINT #1,CHR$(wert);
NSO NEXT
7r  CLOSE 1
ku  KILL "DF0:c/VirusControl.info"
ca  NAME "DF0:s/startup-sequence" A
    S "DF0:s/Oldss"
ZQ  OPEN "DF0:s/Oldss" FOR INPUT AS
     #1
ut  OPEN "DF0:s/startup-sequence" F
    OR OUTPUT AS #2
ip  PRINT #2,"VirusControl"
qq  WHILE NOT EOF(1)
M82   LINE INPUT #1,a$
UW    PRINT #2,a$
th0 WEND
Sv  CLOSE
Ns  KILL "DF0:s/startup-sequence.in
    fo"
I1  KILL "DF0:s/Oldss"
KU  PRINT "Booten Sie nun von der D
    iskette in DF0: !"
HC  END
Ts  Fehler:
IM2   PRINT "Fehler: "ERR
eh    ON ERROR GOTO 0
LGO END
Jx  Werte:
zd  DATA 13592
OQ  DATA 00,00,03,f3,00,00,00,00,00
xY  DATA 00,00,03,00,00,00,00,00,00
gt  DATA 00,02,00,00,00,d8,00,00,0c
F8  DATA 5f,00,00,1a,3d,00,00,03,e9
fo  DATA 00,00,00,d8,48,e7,ff,fe,4b
MC  DATA fa,03,38,41,fa,ff,f2,22,50
CK  DATA d3,c9,d3,c9,24,51,58,89,2b
3I  DATA 49,00,14,d5,ca,d5,ca,58,8a
xD  DATA 2b,4a,00,18,48,e7,ff,fe,61
Lc  DATA 00,01,bc,4c,df,7f,ff,22,6d
uz  DATA 00,14,51,89,20,11,2c,78,00
x4  DATA 04,4e,ae,ff,2e,26,6d,00,18
uf  DATA 50,8b,20,1b,2e,00,e7,88,22
KO  DATA 3c,00,01,00,00,4e,ae,ff,3a
cm  DATA 2b,40,00,0c,50,8b,20,07,53
GI  DATA 80,22,6d,00,0c,22,1b,58,89
Fw  DATA 54,81,22,c1,51,c8,ff,f6,2f
5w  DATA 0b,2c,07,53,86,26,6d,00,0c
Pz  DATA 28,4b,4a,9b,22,3c,00,01,00
y2  DATA 00,20,1b,58,8b,08,00,00,1e
H2  DATA 67,04,08,c1,00,01,e5,88,4e
```

Listing. »VirusControl« macht Schluß mit Viren aller Art ▶



```
61 wt  DATA ae,ff,3a,28,c0,58,8c,51,ce
62 xU  DATA ff,e0,26,5f,7c,00,7a,00,0c
63 Ji  DATA 6b,03,e9,00,02,67,00,00,9a
64 5Q  DATA 0c,6b,03,ea,00,02,67,00,00
65 NT  DATA 90,0c,6b,03,eb,00,02,67,00
66 JH  DATA 00,a8,0c,6b,03,ec,00,02,67
67 QK  DATA 00,00,b0,0c,6b,03,f2,00,02
68 ii  DATA 4e,71,4a,9b,7a,00,52,86,be
69 YP  DATA 86,6e,c6,2f,07,43,fa,ff,18
70 Ry  DATA 20,6d,00,0c,20,10,58,80,e4
71 Jq  DATA 88,22,80,52,87,53,87,67,1a
72 Uy  DATA 22,50,20,28,00,04,e5,88,22
73 Os  DATA 80,20,28,00,08,58,80,e4,88
74 Oo  DATA 23,40,00,04,50,88,60,e2,2e
75 tk  DATA 1f,42,a9,00,04,43,fa,00,2a
76 Pu  DATA 20,6d,00,0c,20,50,50,88,22
77 tW  DATA 88,22,6d,00,0c,20,07,e7,88
78 DM  DATA 4e,ae,ff,2e,22,6d,00,18,51
79 mf  DATA 89,20,11,4e,ae,ff,2e,4c,df
80 Yg  DATA 7f,ff,4e,f9,00,00,00,00,61
81 Lj  DATA 28,4a,9b,20,1b,e5,88,20,4b
82 bx  DATA d7,c0,22,6d,00,0c,22,06,e7
83 a9  DATA 89,22,71,18,00,50,89,4e,ae
84 gd  DATA fd,90,60,00,ff,3e,61,06,50
85 lj  DATA 8b,60,00,ff,36,4a,85,67,02
86 Qa  DATA 52,86,7a,01,4e,75,4a,9b,20
87 Y1  DATA 6d,00,0c,20,06,e7,88,22,70
88 ZS  DATA 08,00,50,89,20,1b,67,24,22
89 uB  DATA 1b,e7,89,20,6d,00,0c,28,70
90 Qc  DATA 18,00,50,8c,53,80,20,4c,24
91 Jt  DATA 49,22,1b,d5,c1,22,12,d1,c1
92 Rv  DATA 24,88,51,c8,ff,f0,60,d8,60
93 OT  DATA 00,fe,f0,48,e7,ff,fe,4b,fa
94 mB  DATA 01,80,41,fa,ff,f2,22,50,d3
95 8f  DATA c9,d3,c9,58,89,2b,49,00,14
96 28  DATA 2b,7c,00,07,00,00,00,18,61
97 qD  DATA 0c,4e,b9,00,00,00,00,4c,df
98 Te  DATA 7f,ff,4e,75,20,6d,00,14,2a
99 rQ  DATA 18,22,6d,00,18,d3,c5,d1,d0
100 3o DATA 91,fc,00,00,00,04,4a,60,6a
101 5Q DATA 02,53,88,10,20,e3,08,66,04
102 29 DATA 10,20,e3,10,64,5a,42,41,e3
103 du DATA 08,66,04,10,20,e3,10,64,42
104 yc DATA 47,fa,00,38,76,03,42,41,14
105 ud DATA 33,30,00,48,82,78,ff,e5,6c
106 IS DATA 46,44,53,42,e3,08,66,04,10
107 ei DATA 20,e3,10,e3,51,51,ca,ff,f4
108 at DATA 4a,43,67,06,b8,41,56,cb,ff
109 mW DATA da,14,33,30,04,48,82,d2,42
110 mo DATA 60,00,00,0a,0a,03,02,02,0e
111 uG DATA 07,04,01,13,20,33,c8,00,df
112 W1 DATA f1,80,51,c9,ff,f6,47,fa,00
113 De DATA d6,26,6b,00,14,50,8b,b1,cb
114 zd DATA 6f,00,00,c8,47,fa,00,3a,74
115 KN DATA 03,e3,08,66,04,10,20,e3,10
116 f7 DATA 64,04,51,ca,ff,f4,42,41,52
117 nr DATA 42,16,33,20,00,67,12,48,83
118 Pv DATA 53,43,e3,08,66,04,10,20,e3
119 ge DATA 10,e3,51,51,cb,ff,f4,16,33
120 RV DATA 20,05,48,83,d2,43,60,00,00
121 gw DATA 0c,0a,02,01,00,00,0a,06,04
122 Jz DATA 03,02,0c,41,00,02,67,42,47
123 zc DATA fa,00,34,76,01,e3,08,66,04
124 Du DATA 10,20,e3,10,64,04,51,cb,ff
125 OT DATA f4,52,43,42,42,18,33,30,00
126 Ak DATA 48,84,e3,08,66,04,10,20,e3
127 tN DATA 10,e3,52,51,cc,ff,f4,e3,4b
128 Ru DATA d4,73,30,04,60,00,00,32,0b
129 PU DATA 04,07,00,01,20,00,00,00,20
130 Lk DATA 00,00,42,42,76,05,42,44,e3
131 1K DATA 08,66,04,10,20,e3,10,64,04
132 EC DATA 76,08,78,40,e3,08,66,04,10
133 FG DATA 20,e3,10,e3,52,51,cb,ff,f4
134 Ze DATA d4,44,45,f1,20,00,48,c1,d5
135 ZO DATA c1,53,41,13,22,51,c9,ff,fc
136 12 DATA 60,00,fe,c8,4e,75,00,00,00
137 Jr DATA 00,00,00,00,00,00,00,00,00
138 Ks DATA 00,00,00,00,00,00,00,00,00
139 Lt DATA 00,00,00,00,00,00,00,00,00
140 o4 DATA 00,00,a0,5a,00,00,03,ea,00
141 MF DATA 00,0c,5f,00,00,68,f4,00,00
142 zs DATA 31,7c,b2,f3,1a,cc,02,7d,c4
143 2P DATA 73,45,29,28,a6,a6,03,82,00
144 14 DATA 19,f1,34,33,cc,48,f9,01,ad
145 Li DATA 23,26,a6,f6,de,40,4a,a9,00
146 OE DATA ac,66,4e,2f,00,79,fe,61,80
147 S6 DATA 22,5f,41,e9,00,5c,ee,f7,8c
148 DX DATA d1,04,be,24,40,24,6a,00,dd
149 61 DATA 13,12,23,ea,a8,03,70,31,ba
150 0b DATA b8,1e,d5,a6,82,46,90,dc,83
151 DB DATA 68,35,03,ed,73,e2,54,97,20
152 HE DATA 60,12,7b,1f,fa,d0,ab,c4,d6
153 yV DATA d7,b6,ed,14,2c,a4,f9,fc,9b
154 Ww DATA c0,8d,33,b8,34,dc,66,28,75
155 yd DATA 9f,d0,aa,c1,09,1e,d8,64,92
156 qY DATA b9,0c,38,2e,17,c4,48,9d,17
157 S1 DATA d0,b3,01,38,23,4d,73,e4,88
158 wi DATA 69,cc,0e,c7,19,72,59,0e,07
159 po DATA e0,67,1c,99,31,02,86,b3,46
160 Me DATA 6d,75,e0,77,85,6e,1c,72,6e
161 oi DATA 22,b5,79,c0,30,7b,cd,a8,dd
162 xH DATA 26,6a,f3,80,5c,f7,9a,8a,cb
163 Oe DATA 02,2a,67,a4,46,60,c0,88,1b
164 pM DATA 21,6c,41,ee,01,5e,43,fa,08
165 SW DATA 78,cf,1f,30,ec,20,40,41,eb
166 hY DATA e5,bf,23,c8,6c,34,e0,0c,90
167 a3 DATA e0,26,64,18,98,7b,e7,2c,d6
168 uZ DATA 72,d6,10,b4,59,73,66,a9,be
169 aE DATA 34,42,10,0c,26,7a,16,d8,f4
170 nO DATA cd,1c,96,5d,66,4a,0f,8e,26
171 k2 DATA ea,c8,70,7f,b1,8b,5b,df,a6
172 rP DATA ea,a4,f8,ac,2a,dc,0c,a5,06
173 59 DATA 48,99,00,18,d9,76,04,86,4a
174 Ds DATA b6,a3,7c,66,7d,88,91,08,98
175 IK DATA 67,52,c1,c5,67,66,08,9c,0d
176 KY DATA 30,06,60,00,16,ac,fa,a5,88
177 uU DATA 22,3a,07,d6,67,04,3e,f9,8c
178 no DATA 2d,9a,b3,2f,ec,03,83,fa,2c
179 HA DATA 8f,ce,14,6d,48,6a,16,0c,20
180 HS DATA 2d,3b,ac,58,a6,aa,41,81,60,6a
181 I8 DATA 94,9f,9c,7c,d5,07,a8,80,bf
182 2x DATA 0c,86,22,7a,32,be,20,09,7d
183 Wx DATA 33,74,52,28,03,77,41,fa,35
184 PT DATA e8,64,00,5a,00,4d,ab,00,69
185 LT DATA 18,02,ac,b9,b0,02,06,1c,54
186 Sn DATA 01,52,db,60,00,6d,86,00,35
187 k6 DATA 83,00,ba,1c,b6,d4,a0,ee,26
188 zM DATA ea,35,a8,e0,01,58,eb,60,00
189 YH DATA 4d,8b,00,35,84,01,56,1c,54
190 Zh DATA 2a,01,ac,91,b0,01,ae,1c,54
191 iu DATA 15,02,ed,02,a2,25,02,b4,9e
192 rr DATA 02,a6,02,d0,48,03,9a,04,03
193 Ha DATA 6b,35,03,ad,03,a2,0d,03,b4
194 nQ DATA 8e,03,46,04,d0,ba,04,32,58
195 oG DATA 0e,2a,98,05,06,05,d3,4e,05
196 ct DATA 9a,06,42,4b,06,69,06,04,82
197 F9 DATA 2c,07,55,04,44,fb,04,68,11
198 hx DATA 04,fe,39,a8,28,05,5a,06,42
199 or DATA a3,61,06,2d,06,e6,26,07,15
200 5S DATA 03,07,6b,36,07,8a,2c,07,5e
201 Zv DATA 35,9e,a8,41,f5,c1,07,1a,08
202 vb DATA 4c,13,08,69,17,08,6d,08,6a
203 qK DATA 35,09,10,39,a8,34,09,9a,02
204 zm DATA 09,60,cb,41,8c,0d,2a,0b,09
205 pL DATA d6,24,09,b0,58,0e,dc,aa,09
206 DU DATA 80,69,b0,0a,86,c1,0a,00,72
207 FB DATA 2c,50,83,28,1a,54,1a,0a,50
208 eR DATA 2c,07,7c,35,78,a8,41,25,c2
209 1m DATA 0a,a0,72,cc,50,83,c8,1a,f4
210 SH DATA 54,0a,01,d3,0b,68,05,0b,18
211 Im DATA 39,44,a8,41,25,0b,d2,0c,0b
212 7d DATA 5a,04,0b,6b,14,0b,ac,a9,0b
213 JC DATA b4,83,0b,b8,1c,e4,54,0b,01
214 03 DATA 93,0c,69,08,0c,08,39,1c,a8
215 AV DATA 41,75,10,3e,90,83,48,1a,4c
216 7F DATA d4,a0,7a,10,68,0f,10,8e,39
217 v9 DATA a8,12,10,9c,72,50,59,10,34
218 9t DATA 8b,10,46,10,d0,20,10,ec,72
219 Yx DATA 50,75,11,b4,8e,11,b6,11,d0
220 1J DATA 2e,11,3c,72,f0,52,83,4a,10
221 4R DATA 0a,21,07,c4,35,a8,14,0f,de
222 VF DATA 72,50,cd,0f,34,85,0f,b2,1c
223 K7 DATA 54,35,0f,4d,0f,86,21,07,40
224 OI DATA 35,a8,14,0f,5a,72,14,50,83
225 Ug DATA 4a,0f,2e,21,07,e8,b5,a8,14
226 UO DATA 0f,02,72,bc,50,83,4a,0e,a1
227 xP DATA ad,0e,90,e4,a0,52,0e,aa,c8
228 Et DATA 41,85,0e,d1,14,0e,7e,72,38
229 ia DATA 50,83,4a,1a,52,d4,a0,0c,06
230 nJ DATA 1e,55,0e,26,80,83,e0,1a,f2
231 w7 DATA d4,a0,fa,06,55,0d,45,ab,0d
232 oH DATA ce,c8,41,d5,0d,d0,04,0d,a2
233 pU DATA 72,5c,50,83,4a,0d,76,20,07
234 pB DATA 30,55,0d,04,80,83,d8,1a,ac
235 E4 DATA d4,a0,5a,0c,92,c8,41,9a,0d
236 uy DATA 54,6a,66,50,83,6e,1a,3a,54
237 Kk DATA 0c,42,01,0e,64,6a,68,50,83
238 W1 DATA 74,1a,84,d4,a0,88,06,94,35
239 2M DATA 98,a8,41,a4,0d,a8,6a,b4,50
240 P5 DATA 05,11,c6,38,6c,a8,02,12,80
241 Ft DATA 1c,a6,d4,a0,da,06,d6,55,1d
242 It DATA d2,80,83,1a,1a,22,d4,a0,36
243 oK DATA d1,a8,fd,7e,d6,0b,d2,cb,62
244 UE DATA 56,97,12,ea,e8,be,05,7a,50
245 bJ DATA 48,00,6c,f8,df,20,09,22,4e
246 QD DATA 20,fc,22,fe,38,2c,2f,5c,52
247 7y DATA e6,99,4d,c0,6a,6a,2f,56,3f
248 s2 DATA ae,ee,9c,6a,3f,b8,99,61,75
249 rJ DATA 02,9e,26,88,df,3f,e0,66,2e
250 sY DATA 99,40,20,d9,6b,c7,75,ea,d2
251 93 DATA 7a,b6,ef,f4,ca,40,10,06,38
252 tA DATA 13,67,54,47,ea,2f,d0,0c,58
253 I3 DATA 47,11,67,06,55,88,24,d8,60
254 6x DATA 0c,30,18,5c,c0,7f,c9,24,d0
255 8P DATA 31,95,58,88,b5,cb,65,e4,30
256 T5 DATA 3c,12,6b,20,49,f9,c3,f0,44
257 GD DATA 2c,42,18,ec,26,3d,a3,2e,ce
258 dG DATA 18,57,c5,34,5a,15,49,ab,6f
259 VM DATA cc,43,d0,ec,b0,09,c0,97,cb
260 Lu DATA b3,a7,c6,e6,aa,41,81,60,6a
261 G1 DATA 67,44,4f,49,4f,06,8f,6a,98
262 x3 DATA 2b,8e,97,8d,00,d9,cc,02,0f
263 rw DATA 56,74,36,c0,40,e8,32,e4,c3
264 LO DATA 60,42,0d,2b,64,20,36,04,e4
265 ps DATA 32,1f,a2,15,c1,72,52,07,28
266 U4 DATA aa,31,9b,1d,9a,15,41,54,ce
267 1D DATA 8e,9e,23,fc,36,ca,20,ad,06
268 BO DATA 0a,a2,13,fc,4e,1f,33,61,02
269 fy DATA 60,66,0e,cf,ac,a0,5d,fa,67
270 XD DATA 32,9b,56,00,0c,6c,a1,d0,ab
271 qA DATA 11,c2,67,46,ba,ed,e8,9a,da
272 pC DATA 4a,c8,39,92,76,01,b3,cb,f2
273 Bi DATA 22,3a,02,ca,ec,75,dc,14,28
274 BN DATA 90,85,97,86,bb,6b,70,c3,7b
275 H3 DATA 7c,d4,1f,00,11,72,79,0b,fa
276 k7 DATA da,b0,bd,28,6a,f8,5a,18,57
277 XU DATA 06,00,86,74,6a,ff,ff,3f,5e
278 bQ DATA 80,1b,09,cc,4e,06,fe,73,a3
279 1c DATA d6,b9,42,9c,96,4c,ca,26,35
280 ir DATA 83,03,a5,02,4a,73,29,86,15
281 3F DATA 85,fb,be,67,3e,76,57,cc,15
282 0Q DATA 7e,01,67,d5,45,36,c3,7b,b2
283 zw DATA 9a,00,b2,b7,1e,27,f2,9c,01
284 aQ DATA b2,b7,4d,e6,20,7a,02,12,f9
285 kh DATA 7b,36,20,50,dc,c6,e1,00,d6
286 Nx DATA 99,f3,29,57,84,60,06,4c,fa
287 LS DATA 01,01,01,ec,f9,bd,01,f0,53
288 LC DATA d5,ac,3b,be,6b,f0,66,04,33
289 kC DATA 6f,f6,35,8f,c4,87,57,67,14
290 Gc DATA 63,06,76,68,dd,e0,32,56,dc
291 Uv DATA d8,d4,a6,74,66,0a,42,7e,a7
292 un DATA ?d,c8,75,6f,aa,49,75,5a,28
293 P2 DATA 2e,69,0a,b2,68,01,6d,66,22
294 BX DATA bb,80,74,06,c8,55,c4,ff,fe
295 hD DATA f2,0d,82,c5,21,44,10,63,08
296 m8 DATA 0d,ce,a6,8a,5a,65,d5,69,73
297 nU DATA 1b,cd,fe,9d,74,b6,79,be,73
298 AK DATA 66,12,5d,72,d5,aa,2a,4e,2f
299 tv DATA d6,52,88,f8,a9,5c,0c,18,00
300 uW DATA 66,66,10,42,29,23,77,23,7c
301 MP DATA fd,c9,6c,30,26,e6,4b,46,2f
302 LX DATA 0a,24,7a,01,28,45,ea,1b,8e
303 64 DATA 4e,92,24,5f,5e,00,fe,0f,f4
304 Q5 DATA 32,c5,b1,80,e0,05,c8,72,32
305 nD DATA a1,a1,61,2c,91,de,3c,c5,01
306 QG DATA 75,40,82,3a,dd,79,5e,54,4a
307 wB DATA 70,9c,61,46,47,01,61,24,6d
308 Eo DATA 61,20,70,03,61,1c,4a,b9,23
```

```
309 wD  DATA fc,53,b0,c6,67,fc,d9,eb,e8
310 FS  DATA aa,3a,37,02,52,f3,ce,2e,ad
311 c4  DATA 2e,fe,1e,2f,00,93,c9,eb,a1
312 Cx  DATA 60,f6,ce,54,3a,b8,13,10,3a
313 bO  DATA 9e,d5,ab,bc,52,71,c1,be,b6
314 Xg  DATA 72,00,20,1f,7c,05,44,3d,b8
315 Bo  DATA 6e,35,82,a2,3a,c0,2d,98,d7
316 7c  DATA e5,1f,ad,0e,26,28,6b,8a,23
317 12  DATA 4b,00,28,70,02,33,e0,27,1c
318 93  DATA e1,88,23,40,e0,5d,24,2c,d0
319 VE  DATA 66,6e,ee,72,04,77,a5,97,14
320 vA  DATA 5a,57,8a,bb,e5,09,00,1c,42
321 22  DATA a9,e5,d7,79,31,38,b8,42,d4
322 oD  DATA 64,db,f3,fe,3e,22,7a,00,34
323 mI  DATA d3,f5,97,6e,02,30,4e,ee,fe
324 qQ  DATA 98,7b,60,02,0f,08,c0,03,b6
325 6w  DATA fc,bf,ff,c5,eb,19,b5,cd,b0
326 J0  DATA 6d,c6,6d,76,b3,89,a2,70,8b
327 3B  DATA 68,67,00,04,36,34,58,32,40
328 sr  DATA 47,2f,69,2d,ef,20,28,cc,32
329 dE  DATA 8b,6b,d6,bc,ac,c4,59,0f,b4
330 Nz  DATA 6c,67,2d,73,29,db,58,d6,2d
331 bi  DATA 2d,96,75,cb,84,2c,b3,56,65
332 OH  DATA 62,59,e6,20,2d,20,54,a6,77
333 ot  DATA fa,00,61,26,67,1e,24,40,20
334 5L  DATA 2a,00,3a,04,cd,3f,2c,5d,68
335 fz  DATA 97,37,77,2e,00,00,09,ae,22
336 De  DATA 6e,00,2e,b3,c0,66,02,fb,63
337 DX  DATA 58,8f,60,00,fc,ae,f4,f6,36
338 aE  DATA 43,fa,0f,3c,7b,c1,da,4a,80
339 se  DATA 6e,23,ec,38,d4,59,82,07,e0
340 pv  DATA 07,f0,b2,66,66,7d,13,31,a9
341 3E  DATA c2,86,ea,02,de,ed,cc,ee,67
342 MV  DATA 25,0d,73,71,6d,41,18,29,90
343 2P  DATA 24,8b,71,46,13,e2,40,35,66
344 3x  DATA fc,e4,18,88,b3,7c,24,81,73
345 xG  DATA 6e,45,68,45,6b,fc,4c,6c,2e
346 0e  DATA 53,79,29,77,6d,a2,7c,73,0e
347 2t  DATA 80,00,63,31,61,6b,88,bf,a4
348 nM  DATA 8e,36,e8,dc,6d,2f,52,70,29
349 rM  DATA 42,73,66,b9,32,23,c1,76,65
350 mQ  DATA 36,08,b3,4b,bb,79,98,b5,d2
351 6L  DATA 62,7a,b2,93,8c,2b,9f,2f,39
352 Qk  DATA 97,47,07,70,66,7d,ce,80,ce
353 g6  DATA 14,56,d6,cc,66,65,58,6e,c2
354 ah  DATA 72,ee,8d,ba,e7,59,74,2b,4c
355 FZ  DATA 38,5b,f1,f4,2e,ca,d3,37,12
356 FF  DATA 5c,76,0c,32,2e,64,e4,cb,a3
357 YB  DATA 5a,58,8c,9b,8a,19,9d,dc,24
358 DH  DATA 34,07,78,8a,70,c3,c0,2a,52
359 ea  DATA c2,2d,e7,18,67,8c,6b,2b,11
360 bB  DATA 0f,6e,2b,82,2b,20,5e,05,f0
361 G1  DATA 22,30,18,58,19,b2,c5,3d,c3
362 Gq  DATA d2,21,b2,a8,b4,6e,64,35,4e
363 4z  DATA 64,2c,14,0b,ca,7f,ac,f3,85
364 jW  DATA 59,68,dc,61,75,4c,3d,c2,5d
365 2B  DATA 98,00,5e,33,15,09,f5,20,af
366 7H  DATA 3c,6c,be,c5,e2,6b,2e,60,27
367 ol  DATA 01,65,6f,09,95,72,39,0d,b3
368 ra  DATA 26,c1,28,91,07,29,70,ad,ac
369 qd  DATA 04,d6,f1,f8,9a,e9,68,72,f1
370 VQ  DATA 73,8e,40,34,c5,e7,50,e3,66
371 rx  DATA 21,e3,ce,60,06,30,5f,b0,00
372 Pe  DATA e5,98,73,69,6d,75,6c,cc,e3
373 AO  DATA f0,42,1e,09,02,3c,a2,40,b4
374 PD  DATA 1f,78,80,07,23,37,0d,0c,45
375 rB  DATA 20,96,0b,34,cc,3c,b0,30,38
376 ub  DATA c7,c3,7c,62,6c,6f,e7,16,f5
377 Rk  DATA 85,08,20,ab,93,c9,88,36,dc
378 bv  DATA 16,e0,00,9e,10,69,0c,9f,70
379 W1  DATA 8d,22,dc,17,b9,53,03,9c,df
380 13  DATA ae,15,2c,44,34,57,64,32,71
381 Gb  DATA ae,04,ce,53,4d,75,91,34,2d
382 y9  DATA 4b,e2,a7,75,5d,7d,24,86,f7
383 CG  DATA 10,2b,d9,2d,41,0e,de,11,60
384 1Z  DATA 72,4a,07,4b,9d,f8,61,20,4b
385 1j  DATA e6,6c,6c,f9,3b,91,71,db,07
386 QN  DATA c0,23,91,1e,8a,ea,5a,88,03
387 TP  DATA 66,9c,4f,52,4b,42,72,b9,43
388 cA  DATA 48,71,9f,d0,c0,6d,f6,67,db
389 Mt  DATA c9,18,76,6f,cd,0a,7a,20,eb
390 je  DATA e4,48,62,79,15,34,72,71,c3
391 Ts  DATA 89,6e,c9,7b,b2,a7,03,19,20
392 mn  DATA 54,2c,d7,c1,54,59,50,45,fa
393 Ai  DATA de,a0,2e,ce,75,4c,49,71,7c
394 5N  DATA ac,18,50,4d,eb,a2,98,3a,29
395 0b  DATA d2,6d,6d,65,6f,b8,b1,c1,8d
396 v0  DATA 4e,fa,e9,53,44,57,bf,6b,89
397 Lg  DATA 6e,2e,24,5e,a4,62,b8,c5,90
398 YY  DATA 11,a5,4e,6f,5c,09,2b,6e,c2
399 KQ  DATA a0,53,3a,5e,16,83,4e,61,6d
400 vX  DATA bb,b9,48,1e,6e,04,c6,73,75
401 Nj  DATA b9,08,96,83,63,d6,89,53,a0
402 2G  DATA ae,2d,53,4c,30,4e,6c,92,66
403 DK  DATA 78,18,c0,ea,d1,65,b5,85,a4
404 Qc  DATA 57,14,64,6f,77,66,95,8b,48
405 Jy  DATA 99,38,6f,68,35,e0,2d,5a,75
406 ia  DATA 67,72,69,66,66,20,f9,21,74
407 Ms  DATA 66,6c,82,28,52,65,71,75,9e
408 3w  DATA c1,03,6b,8b,5c,82,6f,32,1b
409 lg  DATA 35,18,38,a8,4e,f4,96,15,b7
410 uh  DATA 64,75,ef,b9,65,b2,c0,d2,24
411 T4  DATA 53,19,80,cb,86,45,80,11,8c
412 yx  DATA 69,8c,3a,c6,72,64,8b,0f,61
413 jC  DATA 79,52,fc,47,96,66,72,16,f7
414 vJ  DATA cb,97,54,fc,74,7a,b1,bd,4c
415 9T  DATA 47,7c,60,55,69,70,8c,91,4c
416 aY  DATA c5,86,46,4c,cc,c5,4c,9b,b0
417 3R  DATA 6c,9e,4c,2d,66,8b,3d,e4,0e
418 s9  DATA bc,bf,21,c9,71,3d,8e,60,03
419 oY  DATA 74,2c,9e,49,35,29,58,e2,3c
420 q7  DATA eb,a7,e3,ca,d5,8e,0b,8f,62
421 1m  DATA 6f,67,5d,94,2b,8f,64,75,72
422 ze  DATA 8d,94,8b,c9,66,e0,dc,48,78
423 1J  DATA 26,04,6e,7a,8c,7f,20,31,30
424 o4  DATA 30,20,25,7a,06,c1,a7,06,58
425 E4  DATA 4a,2d,45,6e,74,66,70,1e,43
426 v4  DATA 6e,2d,8c,74,68,6f,a0,9e,90
427 fr  DATA 60,69,3a,0a,41,4d,49,47,41
428 EV  DATA f1,7d,7f,66,57,20,b7,21,41
429 eg  DATA 6d,69,67,78,ad,61,6e,d8,55
430 nP  DATA 34,11,f0,64,e9,65,6e,83,6b
431 Hp  DATA 00,25,01,1d,df,ce,88,1e,3c
432 8k  DATA c1,9d,49,ed,8c,69,c1,13,d9
433 lz  DATA 19,e2,ce,6b,53,75,6d,5e,ae
434 TX  DATA 6b,fa,c0,33,33,6d,61,5c,9d
435 MQ  DATA ed,ea,2d,56,65,6b,74,6f,72
436 P5  DATA 20,76,e6,1f,e4,6e,60,63,81
437 Ks  DATA 2d,3e,21,63,69,72,75,f4,3f
438 1N  DATA 0a,9b,33,31,6d,4d,ef,78,27
439 Ie  DATA f0,bc,6f,5c,15,80,62,61,72
440 Fa  DATA 2c,20,64,61,f7,bf,6f,72,65
441 rU  DATA 69,74,73,79,08,1c,69,6e,e7
442 bi  DATA d9,6c,6c,69,65,72,74,20,21
443 1E  DATA 0a,db,fc,5a,b6,4e,03,58,ce
444 4N  DATA 34,c4,09,22,7a,23,a0,bb,3d
445 Yb  DATA cd,97,24,ac,cd,33,d3,99,65
446 mY  DATA 52,50,fd,39,8b,4a,35,18,c0
447 z2  DATA 03,e2,00,00,98,42,58,34,3f
448 vI  DATA 52,0f,87,6a,45,92,d7,f6,20
449 Yh  DATA d2,26,da,76,17,19,1c,22,67
450 vU  DATA 6b,05,16,31,1c,8d,7b,81,77
451 OV  DATA 7c,13,43,3f,60,06,ba,30,cb
452 R3  DATA fd,22,f2,38,f7,38,d3,83,be
453 Hh  DATA fd,51,68,2d,98,74,72,61,63
454 wm  DATA 66,24,12,1f,6a,6b,64,69,73
455 Z1  DATA 66,1c,bd,cb,6b,2e,64,65,66
456 31  DATA 14,4a,7c,2c,0b,03,61,c9,d0
457 5p  DATA 0a,b8,2d,b4,36,69,29,06,6e
458 Vd  DATA f0,1e,1c,c0,b8,3b,08,3c,6b
459 CE  DATA 1f,bc,aa,fb,07,e7,a2,ee,c1
460 UL  DATA 01,96,bc,43,e9,42,13,fb,82
461 S6  DATA 79,4a,13,1d,05,86,16,cb,83
462 eC  DATA 3e,d2,9a,4d,3f,0e,45,3a,5d
463 Q2  DATA 0c,92,6f,0e,82,04,20,92,26
464 LY  DATA 92,58,88,58,8b,3b,7f,a9,de
465 3b  DATA 0c,66,c3,e8,5a,14,43,91,22
466 Tn  DATA 80,45,e8,ff,e4,26,8a,0c,79
467 dn  DATA a6,7f,42,52,66,16,b5,15,7e
468 4I  DATA 9a,83,97,43,1a,43,06,23,09
469 gG  DATA 20,0a,58,25,1f,c2,66,0e,ae
470 OM  DATA 6a,1d,92,ac,ce,31,10,4b,43
471 vO  DATA 1c,c0,96,b7,d6,75,40,6d,28
472 Uv  DATA 6a,da,48,86,48,c6,d1,c6,54
473 Xw  DATA 3f,0f,d6,18,80,3c,58,04,c5
474 aK  DATA 52,10,7a,3d,47,04,44,96,9b
475 FZ  DATA 60,30,25,a4,67,fb,b2,22,13
476 Ou  DATA 10,10,42,10,4a,eb,93,58,60
477 UO  DATA bd,3a,8c,05,19,75,21,8c,cb
478 VS  DATA 66,37,03,88,ce,4b,37,4d,36
479 Ys  DATA bb,98,d6,8d,b1,35,15,c6,66
480 vs  DATA 0c,66,7e,d5,a3,6e,18,b0,05
481 pI  DATA 1c,58,02,82,00,d0,ac,a8,2d
482 0e  DATA 04,6a,5b,76,05,08,20,4d,90
483 Ow  DATA fb,da,6c,87,01,f9,99,66,2e
484 zD  DATA b9,ab,f7,66,26,59,36,a0,3e
485 fP  DATA b3,50,98,59,cd,67,66,d6,76
486 J8  DATA 98,8a,cf,66,0e,0c,98,ec,f8
487 sK  DATA 32,66,06,0c,37,c9,6c,cf,ba
488 8W  DATA a3,f0,3f,00,0c,a9,e5,59,65
489 Ez  DATA dc,c2,20,6f,00,3e,b1,d4,1e
490 Aa  DATA 40,9e,65,ac,06,b4,0c,50,dd
491 B7  DATA 13,2b,ac,17,de,20,14,b0,ba
492 5u  DATA 7c,98,57,a1,a0,fc,99,43,95
493 BB  DATA a7,67,73,14,3b,6b,c0,ae,cb
494 6b  DATA 23,90,96,4d,2a,5c,c9,2b,06
495 ai  DATA e5,02,aa,82,02,82,e5,6e,00
496 bf  DATA ef,ae,96,2d,ac,04,5b,97,2b
497 GC  DATA 30,cb,9c,2a,de,63,ce,2b,02
498 Md  DATA f1,1f,5a,ac,ea,30,0f,00,ce
499 Dv  DATA 4d,1c,d4,05,1d,ee,69,0a,ad
500 Li  DATA c7,b1,0a,a6,e3,1d,b2,32,22
501 7a  DATA 1c,83,20,28,24,48,d8,1e,be
502 bf  DATA a1,1c,04,1a,4c,c4,03,f0,1b
503 zM  DATA 36,8c,86,09,cc,44,df,cc,2c
504 Bk  DATA 03,f2,5c,33,d5,b9,5a,4f,1b
505 5I  DATA 48,64,1a,0a,99,05,b7,1c,5a
506 MM  DATA c4,20,f0,e5,35,01,b2,bc,cd
507 6C  DATA 60,b6,cc,01,4e,af,30,8f,40
508 IG  DATA a4,73,2b,1f,c1,22,8b,87,25
509 wk  DATA df,d6,0a,cd,e3,1a,a6,2c,1e
510 wu  DATA ec,c2,64,39,01,c6,33,24,03
511 G0  DATA 6c,9c,87,90,6a,87,1a,8e,d9
512 BS  DATA 07,c6,1a,bc,f9,3e,9b,56,c3
513 vD  DATA 9a,09,d2,99,63,c7,19,fd,1a
514 zw  DATA 50,27,cf,c7,66,ea,c0,03,f0
515 mc  DATA 19,1c,dc,6e,6c,15,76,02,de
516 81  DATA 94,5c,ac,bc,42,32,83,8c,44
517 kh  DATA a5,9f,60,f3,10,2a,bc,07,c2
518 BC  DATA 54,9e,c7,05,f4,16,0c,a8,36
519 wa  DATA 09,cf,ff,fc,67,c4,4a,fc,b4
520 qU  DATA 14,66,be,ba,43,14,6f,ad,62
521 BP  DATA 66,18,88,64,1e,ec,60,ca,c5
522 br  DATA b0,19,e2,c2,ff,2c,09,87,72
523 kL  DATA 23,15,e0,28,c5,e2,02,12,52
524 JB  DATA fd,0e,2b,0d,ce,d6,50,38,80
525 j5  DATA 06,61,0a,66,77,05,61,b1,01
526 1P  DATA f4,60,9d,10,00,f0,cb,d5,30
527 2M  DATA dc,b1,8c,45,5a,16,c0,47,60
528 gI  DATA 0e,68,49,14,64,20,be,91,2b
529 DS  DATA be,02,14,e5,1d,60,7a,a2,e7
530 nq  DATA ee,1e,30,d1,ad,06,0a,60,59
531 8Y  DATA 06,5f,e2,77,10,0f,e0,65,e8
532 VH  DATA 14,4a,39,cf,e3,5e,22,f2,0c
533 Re  DATA 01,9a,f9,0f,ce,71,fc,0a,87
534 fg  DATA 44,24,08,76,01,e5,0d,74,01
535 j3  DATA 9e,19,1e,01,6b,62,e9,be,94
536 Zi  DATA 53,08,6c,cc,8d,bd,cb,70,9a
537 qi  DATA e0,91,c8,b5,35,0e,00,98,8d
538 We  DATA 66,ac,39,02,80,76,64,fb,68
539 oS  DATA a4,2f,2c,5f,f0,18,ee,20,1f
540 Jd  DATA ba,2e,66,2f,24,5f,7f,1d,06
541 Jr  DATA 20,08,20,5f,7b,1c,98,21,dc
542 QL  DATA 3a,78,d8,81,23,c0,67,72,d8
543 Tx  DATA 83,f7,56,55,7b,98,82,27,18
544 qo  DATA 68,c2,12,b1,ee,8a,00,06,24
545 mp  DATA 2f,29,6d,1a,18,3a,0c,6b,21
546 ZN  DATA 55,67,16,99,12,ee,66,0a,df
547 p5  DATA 86,97,ad,0c,d9,80,0c,5b,75
548 nI  DATA e0,08,47,38,c5,04,36,73,c0
549 sl  DATA c6,d1,c0,2a,48,23,48,c0,df
550 qI  DATA d5,9e,ac,13,1a,4c,60,56,51
551 Rq  DATA ea,eb,08,9c,33,92,70,d9,a0
552 Ut  DATA 28,23,5f,00,24,7c,06,69,15
553 oz  DATA 08,28,15,a3,40,67,90,83,6a
554 DL  DATA fe,3e,66,ee,5d,18,3b,03,2a
555 GM  DATA 02,36,ff,7e,71,d7,2c,62,1c
556 xz  DATA 29,a1,06,ff,56,44,d0,98,64
557 ac  DATA 02,52,80,f5,3b,b0,f8,46,bc
```

Listing. »VirusControl« macht Schluß mit Viren aller Art (Fortsetzung)



```
558 2I DATA 46,06,42,a8,7c,1f,e4,2b,f6
559 NC DATA 7c,fb,d2,75,99,7c,b8,30,3c
560 Qh DATA 03,ff,ee,e9,0c,94,52,41,57
561 hE DATA 3a,66,10,c3,3f,20,65,06,b4
562 YG DATA e7,62,7f,65,59,41,bc,00,2e
563 81 DATA 52,88,fc,cd,e4,ac,7d,76,ce
564 Ah DATA 5c,67,0c,75,10,06,ea,29,00
565 Z8 DATA 30,31,1f,81,69,32,26,36,94
566 T6 DATA a7,6a,ed,b1,c4,15,1d,98,32
567 wG DATA 0b,20,f1,99,85,21,b0,cc,42
568 M4 DATA 22,04,66,21,25,d0,33,81,8d
569 vh DATA 9c,10,80,d9,76,61,00,17,34
570 7X DATA 71,6e,cc,2a,c6,39,60,dd,0f
571 Qe DATA 19,63,07,0c,15,07,44,4f,53
572 8v DATA 00,54,8c,16,be,8f,bf,f2,70
573 Bf DATA d6,f6,ba,d1,06,28,e0,39,25
574 Pn DATA 67,85,56,d3,02,ae,58,92,cc
575 Ot DATA 31,d7,2c,cf,6e,4d,75,0e,99
576 yg DATA d7,da,25,a7,99,10,87,4a,e3
577 go DATA 81,66,f8,66,0d,3a,c5,00,a0
578 lb DATA 23,c8,b9,03,04,32,3c,00,28
579 ug DATA 14,9e,0f,b4,b6,22,83,05,67
580 ps DATA f8,51,c9,ff,f0,e1,67,60,d0
581 DQ DATA 13,ae,fd,d8,2c,40,f2,14,00
582 bh DATA 46,23,ee,9b,47,67,80,42,28
583 Iw DATA 07,88,88,33,fc,80,81,3b,df
584 Ot DATA f0,96,22,c8,65,24,8d,fe,62
585 Ym DATA 6e,b6,aa,43,fa,00,18,75,cf
586 Ec DATA 80,a0,b5,03,0a,20,96,e9,6a
587 r0 DATA 16,70,66,93,75,70,ff,60,fa
588 WC DATA 64,6f,f9,47,00,67,de,70,68
589 F4 DATA 04,5e,73,2e,11,5b,62,72,61
590 Y1 DATA 72,79,88,ff,83,80,00,0f,ed
591 ml DATA 9b,5e,da,93,e6,80,34,77,49
592 MJ DATA 09,77,6f,0c,6c,69,6e,6b,18
593 2T DATA 47,4d,30,45,61,73,00,7f,20
594 jq DATA 6e,69,f4,74,20,09,c3,64,72
595 2u DATA 75,65,63,6b,74,f8,16,77,69
596 uH DATA 72,64,20,62,ea,71,42,6f,6f
597 Xw DATA 70,7f,b8,61,75,74,6f,6d,61
598 qV DATA 74,69,c1,df,20,46,41,53,54
599 uW DATA 4d,45,4d,14,ff,a7,62,67,65
600 FN DATA 73,63,68,61,6c,74,65,74,f6
601 t0 DATA 47,56,49,52,aa,29,2d,43,4f
602 Bb DATA 4e,54,52,fa,1a,56,31,2e,33
603 JA DATA 20,76,6f,6e,20,d9,fc,d4,3b
604 3i DATA 20,4e,49,50,50,47,45,4e,e3
605 gT DATA 7f,36,36,35,37,20,47,65,72
606 Eg DATA 73,68,65,69,6d,20,38,2c,20
607 AQ DATA 30,36,38,34,33,2f,31,35,32
608 tm DATA 32,00,0c,70,e0,3f,39,00,bf
609 Xx DATA ec,01,67,54,80,cf,7e,20,41
610 Am DATA 0c,10,00,2a,73,cf,7e,67,44
611 tw DATA 33,4b,74,fe,99,a7,66,fa,92
612 d9 DATA 07,f0,ce,22,b5,32,7a,1a,50
613 tP DATA cd,a1,eb,30,69,19,e3,52,10
614 bg DATA dc,41,11,0a,53,fa,f5,56,5e
615 fg DATA dd,20,00,fd,ac,d7,02,d4,04
616 YP DATA c8,06,00,28,23,96,78,67,f6
617 Ox DATA b4,19,1a,12,20,fa,5b,14,8c
618 2W DATA 97,0b,ae,69,09,9e,2d,0c,a1
619 8t DATA b9,1a,b9,ec,03,d2,b2,2e,ce
620 3o DATA 5b,16,30,2a,70,e4,20,16,1f
621 OR DATA 26,d4,ba,58,79,f1,ee,2e,b9
622 91 DATA f6,62,58,2e,ae,82,87,1c,20
623 zN DATA 8a,d1,ed,70,31,60,42,90,77
624 US DATA 28,31,0b,93,48,2c,7a,19,b4
625 qd DATA f8,a3,28,b1,cc,e4,67,3a,d0
626 Aw DATA 24,ff,7c,30,5d,4e,49,6b,d8
627 xH DATA ee,ff,e6,21,4c,d0,1e,21,6e
628 ee DATA 81,35,c8,dd,c2,a7,99,94,88
629 PU DATA 75,60,fe,f8,3d,7c,4e,f9,ff
630 xQ DATA e2,2d,48,ff,e4,cd,4c,e1,5f
631 Wp DATA 0b,67,24,32,6b,ff,6a,53,f9
632 wq DATA b1,16,b6,c9,03,5a,c5,18,f4
633 GO DATA cb,b0,3c,00,9a,42,37,d5,2e
634 Xs DATA ec,c1,3a,fc,a4,22,4e,20,bb
635 09 DATA 3f,f4,e5,22,0f,5c,14,4c,c6
636 eb DATA 16,bd,18,cb,24,17,ca,95,18
637 zY DATA c9,85,e3,6e,c2,fd,8d,05,db
638 6S DATA e4,09,e2,a3,cf,f6,54,47,16
639 6w DATA b6,ec,f3,eb,42,6c,cc,f3,03
640 E5 DATA f0,19,1a,2c,60,7d,7e,9d,42
641 FW DATA 50,02,5f,1c,15,e8,c7,18,5a
642 fg DATA f9,e7,cf,0a,f5,f9,42,02,75
643 yx DATA 06,fa,42,2a,71,16,fa,60,59
644 Q4 DATA 2c,90,f9,e8,b7,d9,5f,1a,18
645 UK DATA 53,4a,98,66,08,eb,95,6a,42
646 ka DATA 12,60,10,4a,12,66,0c,14,ce
647 zJ DATA df,49,8b,23,c7,92,02,3c,20
648 lc DATA d9,56,6a,b0,f6,cf,fd,18,62
649 dt DATA 91,37,5e,26,90,b5,d0,67,26
650 Bw DATA e8,6f,96,d0,27,65,67,69,df
651 bG DATA b6,6e,49,4f,e9,55,02,04,60
652 Bq DATA 0a,bb,2f,c7,3c,da,30,50,eb
653 xC DATA 60,ae,9f,5f,0d,22,5a,18,2c
654 Pt DATA 18,86,18,28,e1,04,58,bb,0a
655 GP DATA 4d,65,6d,f4,46,00,c0,60,fb
656 24 DATA 26,40,1b,26,b4,0f,74,d1,17
657 bS DATA fa,cb,0a,23,26,74,56,a0,4b
658 n0 DATA 67,50,1f,93,74,72,30,10,15
659 sP DATA 2e,66,18,04,1c,02,b0,0e,d0
660 iz DATA d8,07,cc,9a,60,f4,97,4b,69
661 gG DATA 63,6b,c0,13,68,65,63,d8,24
662 HM DATA 6b,73,75,9d,99,14,fc,4a,db
663 eb DATA b5,be,67,2c,dc,2a,40,0b,a2
664 5G DATA 68,04,9e,5d,b0,c2,0b,81,4c
665 jp DATA 44,d9,00,01,01,30,cf,49,5e
666 4W DATA 1d,f4,2e,75,28,24,7a,17,72
667 Gi DATA 29,4f,17,6e,26,88,b5,c8,67
668 Ua DATA 1a,2e,fc,31,8e,94,43,4f,4f
669 zM DATA a0,3d,fc,a5,90,20,bc,19,20
670 Py DATA c0,9c,5a,a7,c2,2c,30,70,e4
671 rY DATA e3,56,41,32,81,62,12,68,9c
672 32 DATA 0c,5d,80,6e,84,9c,fd,d4,66
673 po DATA e8,60,30,7b,12,4c,41,09,b6
674 tE DATA 45,20,82,01,5c,b7,f6,08,00
675 5L DATA 67,d6,f1,0b,22,3a,88,25,2e
676 eW DATA b2,38,a8,4b,f2,18,31,ea,b2
677 uJ DATA 18,38,09,00,aa,96,1b,be,41
678 09 DATA c8,23,7e,72,30,52,81,7b,07
679 QA DATA 06,95,e6,60,2a,99,47,18,b4
680 Um DATA 1f,18,76,4b,e6,a8,81,21,f6
681 mb DATA f8,c6,15,41,00,0a,29,0f,20
682 Hs DATA da,20,92,e0,c7,bd,68,da,32
683 zB DATA c0,f0,a1,af,04,72,00,5f,9e
684 Dd DATA 9a,0e,50,88,06,81,e5,93,2c
685 gd DATA 1a,24,f2,60,2e,db,01,76,3c
686 oh DATA 26,52,24,a0,95,f7,f8,35,3c
687 GC DATA b7,d8,67,de,f3,82,33,0e,62
688 Id DATA 49,79,4c,a0,fe,00,2b,98,5d
689 qi DATA 0f,b0,b8,2a,61,1e,6b,37,16
690 Ux DATA 4a,11,ed,10,ae,59,09,1d,7a
691 Ty DATA 61,0e,bc,46,06,42,11,b7,67
692 UY DATA 64,9e,c5,21,66,60,03,21,ae
693 Q3 DATA 47,fa,1a,f1,93,b8,f7,b8,7c
694 R9 DATA 08,2b,5c,56,f9,4e,75,18,3f
695 ku DATA 2c,87,ad,c7,b9,00,20,67,ee
696 U9 DATA e4,ee,fe,38,f3,b6,fd,05,78
697 Mk DATA 4d,72,34,5b,ee,f0,45,6e,05
698 dG DATA ce,4a,06,d4,c9,76,f6,98,c3
699 SZ DATA 1d,25,b8,7b,cc,95,ce,4a,e0
700 GP DATA fa,55,c9,0c,06,7c,0e,08,ab
701 eQ DATA 8b,29,3c,25,49,00,12,28,69
702 OE DATA 00,22,28,54,f8,7b,f8,dc,70
703 pq DATA be,8a,ac,3b,70,34,6e,04,62
704 it DATA 75,80,93,a2,71,47,46,66,04
705 Io DATA 2e,00,66,12,fd,1a,58,d9,b5
706 R4 DATA 14,5c,20,7d,0e,f7,3e,cb,51
707 f3 DATA b2,4a,6e,d9,28,c4,14,35,ed
708 mQ DATA 3e,56,70,30,18,32,18,b8,23
709 5t DATA 08,58,a0,00,58,8b,32,76,81
710 Ch DATA 3a,b2,f5,d3,c0,cc,1f,71,2a
711 NU DATA 72,37,37,56,07,f4,eb,08,05
712 BO DATA 00,1f,67,12,bd,95,72,45,99
713 to DATA 5a,18,5e,e4,ba,84,28,db,dc
714 Za DATA 3e,80,ff,2f,0e,6f,19,2c,c8
715 WL DATA 00,6f,76,6e,36,04,20,40,78
716 03 DATA 16,91,15,1c,3a,28,00,7a,4b
717 7V DATA 9c,b3,81,1b,09,67,c4,da,59
718 Xh DATA 13,66,16,3b,1a,c0,ce,ab,80
719 M1 DATA 41,cf,60,a8,cc,51,14,66,10
720 sw DATA 2c,57,e4,21,e0,9e,ba,82,85
721 4d DATA 2d,26,f6,7e,56,5d,e6,82,ec
722 Rr DATA 86,8d,22,bd,fd,25,81,fe,86
723 4r DATA b9,44,da,2c,5f,20,47,f3,f8
724 dz DATA 32,14,38,ce,4a,24,47,20,aa
725 xM DATA dc,67,65,d2,18,72,04,c2,c6
726 Nh DATA d1,c1,d6,37,e1,71,9d,b8,72
727 7B DATA b5,e1,0e,d6,dc,0a,42,b3,28
728 jo DATA 83,10,0a,f1,2c,8e,f7,96,96
729 8m DATA 53,e8,0c,18,e6,64,4a,58,5d
730 Vb DATA 2e,27,9d,8c,13,c4,b3,dd,cd
731 Gu DATA 22,90,0c,78,eb,c8,6a,66,76
732 fX DATA eb,86,60,be,15,93,59,2a,e0
733 rP DATA 7d,ea,74,9e,49,fa,14,a4,ed
734 kR DATA 57,e6,bd,02,66,14,da,d1,00
735 Zq DATA e2,de,ac,b4,b6,cc,d0,29,6a
736 aF DATA a7,00,70,65,48,18,80,08,c6
737 xn DATA e7,a2,98,16,fa,d1,22,90,0a
738 Ho DATA 22,e3,0e,12,0c,25,03,f0,0c
739 kv DATA e2,bc,4d,dd,73,33,47,c0,09
740 hh DATA 0a,73,d9,1e,ad,f1,74,18,ef
741 hM DATA be,fb,66,8f,ff,45,8c,f5,f4
742 Eg DATA 75,66,8c,f3,f2,99,0f,04,66
743 a0 DATA 24,d8,03,92,72,51,fd,dc,b9
744 x7 DATA 0c,35,c0,f6,ac,70,31,d6,a9
745 pq DATA 76,1c,f5,c8,04,cb,fd,10,f0
746 An DATA 05,57,09,84,8b,08,68,f3,07
747 5g DATA 01,cc,2c,a0,f6,b4,96,a7,fa
748 mN DATA a0,60,d0,ee,a7,06,26,aa,77
749 qJ DATA 9f,dc,fa,07,66,94,8b,35,f3
750 YJ DATA b6,06,4e,b0,ca,f5,08,66,1c
751 qo DATA 6b,bb,c3,8c,4e,fc,c0,ac,52
752 co DATA e3,12,7a,ec,59,e3,0e,92,ac
753 cj DATA f6,26,21,7e,0b,a4,19,b0,3a
754 ra DATA e3,86,09,12,ba,67,4d,0d,18
755 pH DATA cc,04,13,22,e2,49,01,e4,bc
756 Ao DATA 71,46,f6,78,29,f3,0a,c4,45
757 Bw DATA b5,4d,b2,32,c8,ad,5f,69,f0
758 Yk DATA 00,0f,13,00,2f,b1,01,6a,96
759 5F DATA 4d,31,51,00,09,56,3c,2b,66
760 G1 DATA c8,8a,93,fa,69,01,dc,35,f2
761 mZ DATA 71,d4,09,42,96,61,65,5d,ce
762 gq DATA 40,c0,24,0c,58,c3,61,66,95
763 vG DATA 83,75,2d,0c,54,50,fb,de,4c
764 kb DATA 9f,e3,11,fc,42,90,7e,1d,5a
765 jy DATA ab,80,12,70,78,0d,ce,05,0b
766 Qs DATA 26,91,37,61,bd,ea,6d,19,c0
767 nH DATA 1e,4b,06,e3,f6,e0,02,5e,34
768 19 DATA 07,ac,f5,b4,f2,4c,ac,37,27
769 OH DATA b8,d4,23,35,fa,24,08,66,7f
770 uT DATA d5,0c,e9,fa,e4,0c,76,46,db
771 VR DATA d9,80,55,43,16,de,c0,0a,b8
772 vs DATA 32,d7,cd,24,4d,0c,90,4e,84
773 WI DATA 66,15,aa,8a,b5,88,67,d9,af
774 xk DATA 68,d9,6e,52,8a,93,66,aa,79
775 aZ DATA a7,c0,d6,68,5e,8a,8a,91,03
776 R6 DATA 00,88,63,6e,da,fd,0a,2a,b7
777 Po DATA f1,d2,38,56,a9,44,33,d3,81
778 BJ DATA 32,17,62,14,29,17,0c,66,4e
779 dz DATA ea,87,07,02,96,2d,e6,d9,42
780 8Z DATA 60,9c,50,0a,2c,2a,bc,46,52
781 Tt DATA 45,45,60,16,20,4c,cb,7f,8f
782 DM DATA 2e,82,11,02,f0,36,57,d2,08
783 LX DATA 28,24,0c,76,20,db,6b,05,60
784 W7 DATA 10,24,3a,10,e0,67,0a,04,82
785 qa DATA 78,fd,48,16,03,06,24,0d,db
786 tA DATA c8,3c,4e,50,16,e0,71,c2,f2
787 r6 DATA ba,dd,b3,20,f1,42,f8,0d,66
788 hz DATA 0c,69,ef,cf,bc,8a,fa,50,c3
789 sx DATA 9c,fa,7c,e2,ef,0e,59,19,04
790 tK DATA fc,58,7b,c6,10,58,19,5c,fc
791 3V DATA 80,66,46,87,99,2e,cd,12,88
792 LO DATA a8,b3,20,0c,98,03,28,40,b3
793 8D DATA 81,ce,3c,76,0a,cc,66,88,18
794 B1 DATA 11,42,e7,67,ed,ee,33,1c,97
795 Rw DATA 42,d4,22,53,56,93,6c,79,17
796 WY DATA 06,0a,2b,8f,8d,c3,a5,2b,9a
797 35 DATA 68,75,40,3e,7b,aa,80,47,38
798 Eq DATA a1,9e,74,34,94,ac,75,99,72
799 6W DATA 01,16,9d,1d,70,0b,2c,01,1e
800 ls DATA 1c,8c,53,84,73,83,71,87,72
801 TF DATA 6b,70,c3,1b,93,98,06,3d,aa
802 iJ DATA 00,a6,2f,02,06,e5,07,61,01
803 3k DATA 04,24,40,5d,ca,a2,9a,b9,8f
804 LO DATA 70,8b,07,ce,22,61,27,08,cc
805 ah DATA 39,78,94,77,1b,cf,0e,20,42
806 Cy DATA 41,e8,f9,00,4d,b3,b6,51,cb
```

```
807 ip  DATA ff,de,3c,d3,5c,db,e4,2f,0a
808 70  DATA 78,ff,cd,0f,77,95,51,72,71
809 IU  DATA 69,2e,28,06,97,6c,9f,c0,20
810 BD  DATA 28,57,20,57,f2,a3,b0,1a,96
811 Rj  DATA c7,ec,b9,0b,72,e0,b1,62,60
812 jC  DATA d0,de,c0,32,0b,b7,e5,fb,04
813 54  DATA 58,16,85,11,78,c5,82,dd,4f
814 vC  DATA b8,35,c2,2d,94,a6,f2,c1,0c
815 PN  DATA e4,99,42,d6,f1,1a,99,42,d6
816 g3  DATA 25,7e,35,87,6a,04,5e,c8,2e
817 ct  DATA ef,72,22,ad,4d,27,ab,4a,0b
818 kC  DATA ec,0c,55,e3,22,0f,1e,30,09
819 g8  DATA e8,10,7b,9c,40,db,b5,10,5e
820 pG  DATA 2c,0d,02,c2,65,c8,0c,95,f2
821 4b  DATA 86,f3,67,23,7b,69,59,95,18
822 TW  DATA f7,e3,6d,03,c4,68,00,c6,0d
823 D9  DATA cc,a0,3b,22,da,26,47,2a,14
824 JS  DATA 33,43,58,0f,f5,67,07,a0,03
825 LI  DATA de,a6,10,0c,a8,5c,46,82,0b
826 YL  DATA 8e,71,56,bd,01,0e,99,69,40
827 m7  DATA d3,6d,e1,01,da,2c,27,74,02
828 cB  DATA b7,38,1a,aa,29,23,b2,ca,72
829 p9  DATA c2,10,4d,24,bb,21,86,02,3c
830 eP  DATA 73,20,80,53,02,05,81,03,24
831 q6  DATA 00,b9,a7,60,14,10,80,fb,04
832 1V  DATA 38,26,2f,b6,dd,96,82,c4,4a
833 w7  DATA 83,06,0b,f0,26,57,d7,c3,24
834 QI  DATA 57,fd,0c,c0,20,bc,a9,1e,7a
835 2P  DATA 78,20,3c,91,76,df,20,fb,4f
836 LS  DATA da,40,59,8a,dd,48,03,e8,66
837 1Q  DATA 1c,f6,a5,32,32,b8,e9,19,be
838 zj  DATA 07,82,78,24,0c,9a,69,a3,d9
839 Dc  DATA 6c,ed,da,d6,27,16,d0,aa,ff
840 Gj  DATA fc,5b,be,09,02,5e,80,c0,d0
841 Jo  DATA d1,90,4a,90,66,02,56,90,60
842 Sc  DATA ba,16,80,ff,45,2a,0d,80,08
843 9Q  DATA b5,55,a8,45,52,9d,4d,a1,d5
844 zu  DATA 08,b4,d8,66,44,60,a8,ff,c4
845 zc  DATA 03,da,66,3c,7a,80,b0,dc,02
846 Qw  DATA aa,2a,20,49,22,5f,20,1f,e5
847 Fo  DATA 3e,dd,0e,e0,2e,1d,d1,48,5b
848 KC  DATA 42,0f,99,38,97,50,bd,de,14
849 QO  DATA 01,52,c4,ee,58,83,c4,21,40
850 RS  DATA 6d,8f,e5,7a,0c,fc,76,a5,de
851 QK  DATA 18,aa,33,e0,0c,80,d9,6f,67
852 z3  DATA a4,c3,81,15,6d,99,65,32,57
853 oO  DATA 15,4a,09,8e,3a,aa,06,03,dc
854 J9  DATA 37,ff,8a,58,56,0f,71,16,0b
855 mq  DATA 26,96,9d,e1,ff,7a,dc,08,d5
856 XP  DATA cb,ba,18,cd,ac,24,b5,cd,de
857 4o  DATA ee,1c,f2,9d,ae,a3,5f,fa,b5
858 id  DATA ac,02,aa,66,2e,71,2f,82,48
859 c0  DATA 20,3a,0c,7a,b0,92,66,d4,cd
860 zW  DATA ff,27,a4,37,8a,ba,eb,96,23
861 OO  DATA 55,0d,a2,ff,26,e2,27,90,fa
862 sc  DATA 78,1e,74,cb,37,b7,03,d2,22
863 7w  DATA 49,15,e9,76,ea,32,b9,d1,a5
864 OP  DATA e1,cd,8b,00,af,82,5a,ca,c8
865 54  DATA 43,de,0c,24,7c,65,03,ee,cb
866 HO  DATA 1d,a4,7b,b3,c4,66,2a,5a,f6
867 2W  DATA a7,14,ea,75,36,05,f0,7a,14
868 I2  DATA c4,96,1f,64,a8,97,91,0c,22
869 oW  DATA 10,f4,8b,ae,b5,ce,58,07,ac
870 ub  DATA 22,00,24,17,26,d8,df,d4,4d
871 oJ  DATA 1d,27,e0,0e,24,59,0a,32,85
872 Sr  DATA b5,fe,86,4a,84,67,1c,78,ff
873 Kw  DATA e8,40,17,e4,32,f1,01,9e,fc
874 5w  DATA 5c,09,23,09,a8,30,f8,8c,5b
875 GJ  DATA de,cd,4b,1f,a5,4b,ff,4f,01
876 My  DATA c9,b6,10,10,42,10,4a,00,66
877 ml  DATA 8a,5f,38,40,ec,2e,16,5e,0f
878 9S  DATA 66,28,60,23,c0,0b,29,39,41
879 b1  DATA ee,bb,d4,4e,82,0c,68,6a,f7
880 NO  DATA 6e,23,ee,99,58,f3,ba,0e,56
881 c6  DATA 62,b5,10,aa,cf,fd,dd,69,98
882 DH  DATA 6b,f7,36,c1,55,15,7d,ee,e1
883 E8  DATA 4e,58,db,aa,b8,56,16,66,e5
884 9o  DATA 34,91,e2,c8,5a,10,17,18,e1
885 ss  DATA 00,a0,0c,45,00,17,66,1e,fa
886 P3  DATA bc,34,0a,b2,46,de,4d,b8,86
887 UQ  DATA 12,9f,87,08,74,5a,06,16,cc
888 K6  DATA e1,09,7c,58,e1,f4,dc,c5,38
889 BB  DATA 04,0f,28,78,8e,9e,55,3c,ed
890 cS  DATA fc,cb,53,e4,60,30,6c,a3,ee
891 Yw  DATA 97,22,60,22,7d,0c,e0,dc,7d
892 iu  DATA a4,60,10,da,00,74,21,ca,94
893 od  DATA 13,e6,85,41,14,22,58,40,14
894 jJ  DATA be,c6,ec,16,b2,07,0f,5c,0e
895 yV  DATA f0,ba,84,12,4e,74,4e,be,a4
896 E5  DATA 14,12,32,60,d0,de,2d,e0,86
897 Iz  DATA 55,0e,3e,e1,a1,c2,59,3d,0e
898 TD  DATA be,c4,1a,eb,1a,d9,8c,27,00
899 GY  DATA 4a,66,be,91,b5,2c,e0,5a,bd
900 gr  DATA 12,ea,48,60,d2,d5,5e,16,f0
901 c0  DATA 40,5e,52,f5,aa,60,b8,6a,4b
902 PC  DATA 05,78,34,ed,26,7c,38,4d,97
903 Be  DATA 14,29,0f,8d,92,9e,32,41,01
904 qN  DATA d6,60,1a,df,b5,06,78,38,a2
905 kW  DATA d8,50,12,59,b3,86,fa,fe,43
906 r0  DATA 29,13,60,c6,ff,b4,68,1e,29
907 d3  DATA e0,9e,95,13,64,60,e9,32,5a
908 YO  DATA 1f,06,2a,60,78,4e,a7,65,64
909 DU  DATA 67,83,2d,33,17,ba,36,fb,b1
910 CD  DATA eb,92,11,46,e5,22,b6,3a,2a
911 QA  DATA 41,bf,ed,06,76,e8,60,9c,68
912 3z  DATA 3b,10,b8,60,96,c8,8d,04,10
913 dF  DATA 5c,3c,47,ea,b5,58,c0,84,1b
914 c1  DATA d6,55,03,02,3c,11,c0,0e,de
915 RI  DATA c2,aa,12,cc,aa,41,bb,26,0f
916 6V  DATA 47,e2,5b,4e,5a,16,40,0b,60
917 aE  DATA 68,0b,ac,4c,3b,e9,11,87,20
918 w8  DATA fc,42,41,44,20,b9,df,81,f8
919 wq  DATA f5,a3,04,56,0f,bc,60,00,fe
920 24  DATA 7c,99,fe,9e,56,9e,dc,a4,ec
921 iA  DATA 84,d8,20,80,ad,c4,76,8d,ac
922 1P  DATA 5b,18,be,b5,9a,75,16,38,28
923 Nx  DATA ae,98,73,ee,9a,97,cb,de,e6
924 KY  DATA 54,18,00,3c,12,5a,32,37,7e
925 WY  DATA 46,e8,08,d2,9d,93,c6,8a,3d
926 LG  DATA 26,34,6e,ba,ab,06,31,e6,8c
927 SX  DATA 42,33,3e,07,f4,71,08,28,33
928 YK  DATA 7c,00,09,00,1c,ad,0f,81,15
929 ZO  DATA 9c,95,84,c6,31,c4,39,80,98
930 ev  DATA 6a,ce,00,3c,f0,58,97,98,41
931 QS  DATA 81,62,16,1c,e0,da,0d,0e,f0
932 TO  DATA d0,12,01,78,97,c6,4a,90,67
933 Y6  DATA 06,20,50,f4,6f,38,29,02,86
934 bA  DATA 05,4f,c2,b9,7d,a2,40,92,71
935 lz  DATA 9a,e9,8c,05,07,a5,29,02,26
936 OJ  DATA 0a,02,2a,42,ae,00,2a,f4,17
937 cq  DATA b0,3a,46,c9,da,9a,b8,d4,9a
938 YR  DATA 59,5f,65,08,20,7a,07,a4,20
939 O1  DATA ba,07,a8,f5,f3,00,f0,88,07
940 Mb  DATA 7a,e8,00,92,45,f8,fb,70,06
941 ec  DATA 80,bf,08,b9,24,64,e6,80,04
942 Pq  DATA 24,d0,8d,a2,81,ea,ea,23,98
943 ZE  DATA 47,fa,23,d4,45,f8,38,58,70
944 w6  DATA 0f,0c,90,e2,a3,c6,64,40,88
945 aX  DATA 03,91,58,8b,60,0a,24,90,26
946 DK  DATA d8,d5,78,7e,6a,99,0c,8d,e0
947 fz  DATA a2,0a,0f,00,33,fc,c0,e2,2f
948 mL  DATA df,f0,9a,7a,3a,a8,38,96,3c
949 ca  DATA 28,90,01,80,20,bc,10,b7,01
950 90  DATA 53,14,1a,e3,cc,20,08,77,bc
951 dg  DATA 20,b9,45,fe,38,cb,b3,b0,5c
952 ma  DATA 12,1c,0a,88,83,1c,d1,cd,61
953 Q1  DATA b2,01,60,42,90,2e,8b,2d,7a
954 DC  DATA 06,5b,2e,41,ee,a0,59,42,40
955 ht  DATA 72,17,d0,58,d4,a7,fc,46,40
956 UK  DATA 30,80,91,57,30,d0,7d,ff,42
957 1Z  DATA 98,8d,00,20,4d,24,48,60,ee
958 OT  DATA ef,c5,60,96,34,61,4e,2e,8d
959 Gy  DATA 1b,03,48,09,be,70,34,c0,c6
960 ya  DATA d5,c0,61,02,e9,47,70,1f,14
961 8Y  DATA d8,51,c8,ff,fc,c0,af,15,e3
962 G7  DATA 28,41,20,41,4e,0d,33,02,52
963 2d  DATA 30,cd,4b,33,80,84,c1,5c,0c
964 8u  DATA 18,36,66,f6,20,6e,00,22,4c
965 k2  DATA 5f,18,d4,61,04,26,4c,de,06
966 yc  DATA ec,e5,1f,28,e5,88,24,40,12
967 Bj  DATA 1a,48,81,53,41,76,bf,66,12
968 Cg  DATA 51,c9,ff,06,4f,13,00,3a,a5
969 fT  DATA 56,1e,4c,60,20,70,ff,39,9d
970 Fk  DATA 40,59,b1,e4,04,19,40,53,81
971 Ug  DATA 69,0a,67,da,06,23,78,bf,5a
972 w7  DATA dc,43,28,e4,70,d5,70,ee,da
973 yC  DATA a4,0c,a2,dd,02,66,6e,b5,15
974 oN  DATA 0c,92,98,f5,66,b6,f8,76,62
975 s1  DATA 10,b7,ca,65,b5,88,8b,33,b0
976 vA  DATA 4c,04,5c,3b,00,14,b0,9a,67
977 lf  DATA 26,db,7b,0e,22,75,1e,36,a0
978 PD  DATA 2f,b2,f2,0e,3e,d0,6f,26,ee
979 ge  DATA 00,8e,1d,6e,a0,08,20,4c,98
980 Eb  DATA 81,0c,8d,c6,df,94,07,e4,44
981 LJ  DATA 36,04,18,60,0f,4c,4d,fc,16
982 Sk  DATA ec,f8,68,c3,e1,b1,48,58,8a
983 Oy  DATA 0d,f8,3d,1b,e7,16,ca,60,1c
984 56  DATA b5,89,ea,30,60,12,b8,13,05
985 J2  DATA c4,60,08,71,a7,c1,74,e9,22
986 nK  DATA 70,d4,37,b1,fd,d8,4c,02,33
987 mM  DATA 4c,98,e1,ba,2c,5b,5d,8c,63
988 pr  DATA dc,22,4e,ee,0a,ee,fe,62,dd
989 B9  DATA 43,11,e5,17,05,0c,d1,c6,42
990 uy  DATA 19,3e,35,16,16,9d,37,8c,23
991 k2  DATA 95,a0,0d,59,3f,e5,b6,09,ab
992 Xa  DATA 80,35,02,04,dc,52,10,0c,10
993 5m  DATA be,7f,66,10,e2,41,5a,04,c4
994 Kj  DATA 71,97,ac,4f,22,01,76,10,13
995 Fn  DATA 05,28,88,83,6e,1a,70,d4,a0
996 ap  DATA 72,32,e8,6d,81,09,bf,80,db
997 ZB  DATA 89,0a,c0,75,a2,e3,99,b2,7c
998 LP  DATA 0f,b8,46,ae,25,f0,20,60,ff
999 dY  DATA 28,9c,35,03,da,26,49,e5,00
1000 3Q  DATA 00,42,9b,38,c1,94,26,ce,61
1001 6k  DATA 0c,2b,00,32,02,12,67,08,c4
1002 Pa  DATA 3f,c9,60,10,ec,36,b2,14,47
1003 Zu  DATA 96,7d,92,5b,2c,03,8c,2c,97
1004 83  DATA c9,8d,c9,62,12,c1,27,10,cc
1005 Yn  DATA 56,32,01,0a,b3,00,7e,d1,ff
1006 1G  DATA 00,e0,36,07,16,bc,7b,1a,6a
1007 wN  DATA cc,d8,c6,62,21,4a,b0,15,84
1008 16  DATA 0e,00,fe,d0,cf,4f,76,e5,8a
1009 p0  DATA 01,78,70,98,20,d6,d8,53,10
1010 14  DATA 66,e2,4c,9e,04,3b,80,2a,0b
1011 fh  DATA 2f,96,49,fe,a0,20,1f,d6,11
1012 fv  DATA 66,06,c4,4c,60,c4,20,6e,6f
1013 yE  DATA 40,45,31,a1,0e,b5,c8,9a,00
1014 Jq  DATA 66,b4,f3,0a,54,45,04,06,13
1015 IB  DATA 07,6e,35,70,a8,41,72,31,59
1016 zh  DATA 00,74,79,35,09,90,bc,4b,80
1017 Dh  DATA 6d,c8,2e,20,40,4a,80,f9,a2
1018 aC  DATA 78,20,28,6d,67,a0,70,42,a8
1019 Jm  DATA 00,40,05,0f,00,fe,7a,67,01
1020 cu  DATA ec,12,26,56,55,ff,58,f1,37
1021 GK  DATA fc,70,76,20,4f,91,ef,07,03
1022 ia  DATA e8,c8,f7,7c,72,19,e0,a4,2a
1023 7J  DATA 40,2c,78,b1,1b,d8,cc,20,6d
1024 Ai  DATA da,57,b1,32,d9,67,12,ae,fd
1025 8N  DATA de,2a,55,bb,fc,c8,df,fd,e6
1026 lx  DATA 5b,2b,fd,e2,f1,b6,75,93,c9
1027 Vj  DATA ee,a4,da,36,68,23,40,ed,46
1028 Yr  DATA 72,9e,d0,03,78,87,9b,30,70
1029 mM  DATA 01,72,00,f2,01,44,47,64,4c
1030 wT  DATA 01,ab,20,23,4b,b6,f7,08,08
1031 Xp  DATA 35,cc,97,24,22,7a,02,fa,7d
1032 Lh  DATA 28,2e,2d,08,6a,33,3e,43,fa
1033 WV  DATA 2c,28,2c,2f,fe,98,28,7a,03
1034 8b  DATA 0a,a7,fe,7e,b1,ec,68,4b,66
1035 PZ  DATA 16,30,83,e8,d2,98,07,39,50
1036 aF  DATA ff,e2,29,f3,24,08,ff,e4,ea
1037 h5  DATA a7,9c,75,26,c4,91,b0,04,00
1038 Iu  DATA 24,0d,7e,2a,e7,0d,75,97,c8
1039 AV  DATA 2c,f9,fa,74,00,9c,55,8e,f9
1040 Iw  DATA f2,c7,40,20,fc,9c,c5,63,86
1041 ZD  DATA ec,58,27,e6,0d,d0,13,1b,c0
1042 3C  DATA e3,59,dc,21,7c,98,8e,67,f0
1043 wm  DATA a2,29,cd,15,1e,1f,ff,56,01
1044 8M  DATA 4e,ee,ff,82,fb,88,57,ad,20
1045 uK  DATA 6e,cc,20,24,48,22,ed,0a,20
1046 3j  DATA 50,0c,91,31,3e,9c,66,f0,22
1047 Fk  DATA 4a,67,4f,e6,04,1e,ff,41,ee
1048 Mu  DATA 01,42,90,77,20,29,57,0a,60
1049 tK  DATA bc,7e,3d,64,02,7e,ff,6d,64
1050 XR  DATA 4f,e0,67,0c,7b,4c,dc,ae,97
1051 jM  DATA 05,ca,67,34,4a,87,67,2c,d1
1052 kT  DATA ef,86,a1,4e,02,bc,9c,a3,11
1053 h2  DATA 0a,44,a7,04,06,6a,e7,24,d6
1054 OL  DATA b9,c9,a6,dc,20,35,01,8e,42
1055 Wn  DATA 10,e4,0e,00,36,7c,78,f6,60
```

Listing. »VirusControl« macht Schluß mit Viren aller Art (Fortsetzung)



```
1056 36 DATA 06,61,05,96,cb,cc,21,00,4a
1057 Q3 DATA 10,67,06,52,83,52,88,60,f6
1058 Jz DATA dc,7f,66,d4,62,12,22,3a,01
1059 FA DATA 60,67,04,fb,9b,42,27,04,70
1060 ma DATA 34,70,56,b6,fc,9c,61,b0,dd
1061 u6 DATA a4,8e,45,41,b0,62,10,24,08
1062 QX DATA 76,01,e2,79,ff,d0,98,04,ce
1063 bt DATA 52,ba,ff,78,f2,b0,09,bf,d7
1064 Nd DATA 0f,7d,d8,0a,0a,71,0e,88,4c
1065 MO DATA 0f,c1,18,39,fe,20,14,77,01
1066 OD DATA 80,e5,62,8c,2e,92,35,92,73
1067 Yk DATA a1,0c,48,22,3c,fc,0e,48,e7
1068 Dk DATA b4,0d,4e,ae,fe,86,4c,df,7f
1069 rz DATA fe,60,de,f4,b7,11,e0,0c,b5
1070 3x DATA 01,00,04,66,56,e0,8b,64,5a
1071 tE DATA 20,a7,2d,2a,1c,16,c8,30,4d
1072 cW DATA 30,66,38,60,1f,6c,68,0d,e0
1073 h8 DATA 60,2a,5c,60,54,67,08,0c,68
1074 KD DATA 00,61,00,06,66,1a,30,28,00
1075 jb DATA 08,02,03,00,fe,07,0c,40,e5
1076 ev DATA 7b,66,0c,2f,0a,45,fa,00,a3
1077 dy DATA 14,bc,00,ff,24,5f,20,08,4e
1078 DE DATA 75,0d,f0,bf,52,a8,4c,0a,5d
1079 Ky DATA 2a,00,7f,6f,13,23,37,64,22
1080 Ep DATA ec,97,81,02,ff,ff,60,60,0c
1081 wH DATA 0f,48,c1,5b,ef,17,78,6e,76
1082 CB DATA 27,90,01,bc,ae,74,75,8c,26
1083 aG DATA e8,b8,2e,6c,69,dd,2a,03,79
1084 eu DATA 00,74,69,f2,06,72,59,dc,d1
1085 TK DATA 70,18,2e,64,65,76,27,d7,9a
1086 69 DATA 1b,55,f1,05,ef,6b,0e,e0,21
1087 E8 DATA 19,ec,ac,00,4b,65,80,17,e8
1088 wX DATA 64,97,4f,ec,38,2c,a5,1e,80
1089 LJ DATA 03,f2,67,03,02,31,37,3c,9b
1090 hp DATA 42,4f,4f,54,42,4c,4f,43,4b
1091 9Q DATA e4,07,90,71,ed,44,00,76,96
1092 FH DATA 0a,54,56,75,d7,03,2e,be,57
1093 oj DATA 3a,31,00,36,34,01,16,32,30
1094 DI DATA 30,2f,70,41,9a,9d,cb,2d,50
1095 s4 DATA 19,b9,23,e1,3a,56,43,70,72
1096 vn DATA 99,ef,6f,6b,6f,58,03,ea,62
1097 Ub DATA 20,a6,62,5c,3f,e5,85,56,e5
1098 oD DATA cc,4c,0e,2b,20,52,f3,76,48
1099 i5 DATA 49,46,54,94,cf,a0,d4,98,12
1100 Mt DATA 8a,f0,72,6f,31,d3,aa,c5,05
1101 JL DATA 81,75,2f,78,0c,00,78,0d,66
1102 KN DATA 6f,e2,49,1a,f7,b9,0b,79,a4
1103 K5 DATA 72,74,c7,3f,b6,5c,66,0b,9e
1104 Q2 DATA 77,65,0b,37,98,8f,d5,3c,3f
1105 S9 DATA 84,6d,61,39,5e,9c,6c,15,8e
1106 NM DATA c6,91,e5,07,bc,bd,58,ed,b5
1107 t9 DATA c4,62,55,4e,d9,31,c4,d6,98
1108 XP DATA 52,58,09,b0,66,ae,0b,c0,8b
1109 fJ DATA 73,6f,65,da,11,36,fb,8d,a5
1110 ga DATA d4,a1,48,e0,8c,eb,10,1d,dc
1111 QR DATA 54,5a,54,59,a7,82,29,00,37
1112 tN DATA f7,94,dc,96,26,62,2d,37,c5
1113 g0 DATA 3b,7b,1c,67,0e,ef,3f,32,e3
1114 XS DATA 38,83,69,9a,38,b5,62,26,76
1115 2D DATA 97,33,50,52,4f,47,52,41,4d
1116 10 DATA 4d,30,7e,50,49,0f,56,a2,35
1117 2B DATA 50,50,47,d2,53,b1,1a,e8,43
1118 DS DATA 4f,50,59,52,49,47,48,54,f9
1119 qA DATA db,4d,41,52,4b,54,26,54,45
1120 dy DATA 85,9f,77,4e,49,4b,4d,c0,80
1121 U9 DATA c2,57,88,c2,26,f6,73,30,85
1122 Ei DATA 2a,37,98,c2,5e,ab,2f,6b,f0
1123 Z2 DATA 44,fa,9c,51,07,92,da,2a,be
1124 UQ DATA 20,4c,39,5f,67,77,cc,64,2d
1125 y7 DATA 55,6e,74,c3,ef,d1,69,65,cd
1126 1C DATA 95,b2,c4,67,02,27,de,6d,0d
1127 NV DATA 1f,2d,57,f3,64,6f,77,4a,4f
1128 Ia DATA c0,4d,dd,2d,45,72,72,6f,72
1129 Jb DATA dc,ef,2b,e0,27,ee,99,12,c7
1130 8m DATA 59,b8,be,47,61,64,6d,20,29
1131 Kk DATA 49,6e,68,e8,7d,74,2d,b1,90
1132 Gw DATA 72,f6,33,69,66,21,3f,e5,b8
1133 aW DATA 10,9e,2b,28,46,e6,45,44,45
1134 Ph DATA 39,9f,c2,07,28,bc,db,73,7b
1135 aB DATA 89,07,69,78,31,6c,a1,ce,b9
1136 Gu DATA de,d1,8b,71,3a,b2,5c,18,f0
1137 OG DATA b4,67,9c,88,94,52,20,8a,25
1138 MQ DATA 6b,5d,13,5c,c0,28,bb,0d,37
1139 vZ DATA 10,10,3c,72,7b,47,3b,4d,0a
1140 se DATA 28,78,b7,ef,16,25,80,87,ca
1141 D8 DATA 80,b7,0d,b7,51,b9,80,27,0d
1142 ZM DATA 6b,78,56,97,4d,c0,1b,6c,73
1143 Dc DATA 14,53,c8,1c,bc,3d,78,29,b7
1144 zs DATA c3,6f,62,83,4b,e3,a2,6f,0a
1145 Bm DATA 4e,55,03,9e,08,dc,1e,bd,29
1146 Ew DATA bb,6d,20,55,45,77,1f,52,28
1147 8u DATA 20,e7,45,4e,20,6d,69,fc,a2
1148 VS DATA 0b,de,94,02,29,de,6d,ba,d5
1149 5c DATA c6,bb,dc,58,80,47,c3,84,b7
1150 5U DATA ad,b5,cf,43,4f,4e,54,b8,ab
1151 14 DATA 4c,ce,e1,18,80,c7,22,03,f0
1152 vv DATA 1a,63,c1,c0,04,2a,72,3c,db
1153 gp DATA d8,6c,ba,08,67,1e,75,6e,8a
1154 oc DATA 5f,60,00,1e,1f,8c,c1,23,40
1155 A3 DATA c0,2a,be,f3,97,e5,c7,2c,28
1156 7y DATA de,19,c1,b8,17,73,b9,04,33
1157 FE DATA 2c,59,c5,02,25,de,51,ba,c1
1158 r0 DATA c3,04,3c,33,1a,37,2d,e0,69
1159 L5 DATA cf,38,d6,59,80,a7,35,e0,3a
1160 58 DATA 2d,d0,0c,ec,18,08,f0,a0,61
1161 Tm DATA 02,2b,64,9e,27,6b,b4,9e,a3
1162 vK DATA 61,68,61,64,2d,66,0e,2c,20
1163 rd DATA 44,86,fb,6e,b5,66,fe,c6,69
1164 z3 DATA de,d1,12,e0,09,2b,9c,99,8b
1165 lw DATA e3,0c,5e,8b,02,2b,de,6d,ba
1166 Vt DATA 3d,d4,38,bc,38,80,47,28,cb
1167 Kd DATA 4c,a9,57,4b,53,27,ce,3f,8d
1168 27 DATA 90,95,4e,b3,08,14,bc,10,05
1169 IU DATA bc,46,bb,7d,38,e0,25,00,e1
1170 gN DATA 00,01,03,3b,40,16,0a,00,5e
1171 QX DATA a5,dd,ce,7b,52,52,4f,52,9d
1172 g2 DATA 1b,6e,66,1f,e9,10,16,8f,ac
1173 zH DATA 62,8b,35,cb,b0,05,ef,14,2e
1174 M4 DATA e0,2c,ce,39,04,41,7b,54,55
1175 oB DATA 4e,47,e0,20,53,43,48,52,3e
1176 Lx DATA 42,5a,55,47,52,49,46,46,20
1177 XK DATA 61,75,66,26,fe,f4,9a,04,dc
1178 hR DATA 85,f6,80,4c,65,73,db,84,77
1179 EE DATA 0e,ce,a2,97,0e,ac,4e,01,2d
1180 Yc DATA 22,cf,a8,33,01,20,2f,9a,44
1181 3D DATA 46,30,6c,e3,77,69,72,6b,ca
1182 hF DATA 5b,45,aa,fc,d7,72,03,1f,63
1183 a0 DATA 72,65,69,bb,7d,46,e4,0e,ce
1184 zM DATA 49,01,62,78,39,60,4a,5c,c4
1185 ky DATA 51,22,f0,7e,5e,4e,45,49,4e
1186 8h DATA db,eb,66,45,28,e0,9a,65,84
1187 Ft DATA 31,2e,33,f1,6e,8d,73,4e,02
1188 Gz DATA 66,25,01,65,47,8c,03,26,de
1189 h7 DATA ae,b9,69,09,2d,da,78,1e,d0
1190 gT DATA 7c,00,f0,52,6f,75,4d,6e,22
1191 40 DATA fd,7c,c0,01,bc,00,2d,da,6e
1192 YK DATA 17,72,2d,53,69,6d,75,6c,61
1193 zy DATA fb,96,7c,93,60,c0,28,fb,80
1194 E1 DATA 6c,f5,88,73,3a,20,ed,7b,61
1195 Rp DATA d9,65,20,17,65,24,80,d8,75
1196 sH DATA 65,6c,ad,35,a7,69,a4,78,91
1197 MX DATA cf,74,cc,84,a4,42,d3,a5,6b
1198 og DATA b3,2e,6a,33,f9,4e,c0,8b,90
1199 NQ DATA 55,26,cb,8c,02,0f,c2,ce,dd
1200 tS DATA 68,84,2e,d8,4c,ba,01,19,78
1201 nV DATA 1a,04,78,0a,10,78,53,c2,e0
1202 m2 DATA 80,67,02,0f,00,1e,ef,2b,39
1203 UK DATA a7,23,37,cc,5e,57,8a,31,65
1204 Hd DATA 6b,88,b5,78,20,15,e3,e4,15
1205 p0 DATA 1a,56,f7,2e,11,3a,d0,20,3f
1206 eg DATA 8c,c3,c4,00,2f,ad,c6,49,c8
1207 Ou DATA d2,ac,17,0d,c3,66,12,56,49
1208 f1 DATA 52,ea,2d,45,4e,54,46,19,7e
1209 Tn DATA 4e,c0,68,45,52,53,55,43,48
1210 w5 DATA f5,4d,46,c5,74,75,72,7a,67
1211 0t DATA 65,66,61,68,9b,1f,01,29,62
1212 yh DATA 65,73,6c,73,ef,22,4d,49,47
1213 y5 DATA 41,f0,17,20,21,21,20,ec,58
1214 ns DATA 03,5e,00,ed,69,70,69,f6,85
1215 AF DATA f5,72,22,57,65,c8,02,2f,c8
1216 MB DATA cf,03,1a,22,0d,d0,49,19,2f
1217 Dw DATA 78,3b,e7,86,b2,74,73,cb,28
1218 a6 DATA fc,fa,48,8a,96,22,b7,1d,e0
1219 QW DATA cd,d9,2d,19,53,75,6e,9e,d2
1220 cJ DATA 55,b6,e7,1a,62,69,c3,04,30
1221 BB DATA bc,71,b5,0a,88,59,4b,e8,d9
1222 5Q DATA 38,8c,55,53,f3,f5,80,68,77
1223 Cv DATA 40,4c,cb,6b,20,13,e0,2d,5a
1224 Ew DATA c2,75,74,7a,f6,f5,80,17,0c
1225 br DATA 20,4b,37,d3,e0,53,79,1b,ab
1226 Zw DATA 8a,32,6d,1b,15,9d,f3,02,f0
1227 FH DATA b8,7e,77,2d,9a,e1,ce,a8,0a
1228 K8 DATA 30,e0,3c,57,59,02,84,3a,e9
1229 Cr DATA 57,23,07,6e,69,6e,67,2d,c5
1230 RL DATA 97,4b,e0,31,1d,96,42,5b,3e
1231 iR DATA bf,20,56,d9,5b,75,73,43,7d
1232 FT DATA 67,72,6f,6c,db,58,03,5e,5b
1233 UM DATA ad,72,0a,b9,02,bc,2a,a1,2a
1234 gn DATA f0,b4,a7,66,2e,2a,a7,f1,73
1235 c3 DATA 75,6d,dd,51,94,e1,cb,b0,06
1236 39 DATA bc,68,5a,15,4e,6f,5c,73,74
1237 LJ DATA 28,46,d1,e5,84,e0,a8,7d,c5
1238 7d DATA 74,61,6e,64,db,b7,64,24,89
1239 kt DATA e7,03,0f,50,aa,9a,c0,1a,08
1240 Mq DATA 81,3a,ec,b2,05,31,f8,3c,03
1241 rC DATA d0,62,05,a0,ab,c6,6c,65,99
1242 FV DATA 7f,e1,bc,00,bc,1a,b8,42,6f
1243 D1 DATA 6f,74,42,79,b7,6b,20,2d,3e
1244 Dm DATA 5c,1f,c2,69,72,6d,b9,23,e0
1245 Vo DATA 07,a2,c9,38,a0,cc,25,48,54
1246 9A DATA ca,20,46,61,72,62,3c,94,1e
1247 0y DATA c0,41,4e,33,a6,9c,93,97,41
1248 xN DATA 78,81,2f,d7,70,66,41,17,dc
1249 Uo DATA 02,5c,b9,09,34,6d,81,2f,b4
1250 y8 DATA e7,01,16,bb,05,88,72,23,60
1251 98 DATA 4d,5b,e0,2f,dc,f9,ba,63,60
1252 JD DATA 00,b4,02,25,ef,e5,05,4f,5f
1253 h6 DATA c0,63,e0,02,9c,b5,15,b1,0f
1254 9n DATA 84,80,a1,d1,0c,13,bc,05,0c
1255 FD DATA 72,2b,68,52,bd,73,a6,07,9a
1256 2p DATA f3,af,bb,12,f0,12,16,c8,38
1257 Bz DATA 00,23,1b,82,ea,00,58,3d,13
1258 DU DATA 4b,05,77,cc,c0,11,5b,20,64
1259 z5 DATA 37,81,06,e1,05,30,de,63,38
1260 S6 DATA 17,54,d8,9f,33,90,73,04,ac
1261 oF DATA 41,78,81,30,2c,e7,01,3b,ab
1262 XC DATA 1f,a7,02,72,54,0c,c2,0b,30
1263 pt DATA bc,e2,3e,00,0e,2f,15,33,e8
1264 JZ DATA e7,04,d2,c3,c0,7c,3b,60,e7
1265 y8 DATA 2a,e3,54,14,4c,a0,3d,0b,30
1266 4R DATA a4,3c,0f,d8,1d,c2,34,40,00
1267 D8 DATA d0,b9,eb,61,e0,30,85,99,01
1268 Nc DATA 0b,ba,05,72,a4,22,70,b3,65
1269 4s DATA c1,88,01,dc,bd,2f,21,26,2b
1270 4B DATA 06,03,f0,f0,60,86,80,b1,00
1271 4J DATA a5,11,ab,c0,30,f4,f3,80,1d
1272 5G DATA 88,18,c0,c4,4d,20,53,0c,3c
1273 4e DATA 1e,03,07,74,a9,34,f0,38,97
1274 dz DATA d3,a9,05,31,de,b7,58,03,72
1275 wI DATA 41,b1,0e,34,c4,51,95,f1,ba
1276 80 DATA c0,53,82,59,10,57,10,eb,40
1277 rX DATA 48,9b,87,40,af,0b,94,bc,03
1278 LX DATA 76,a3,74,a0,74,4d,a0,27,0b
1279 p4 DATA 31,bc,3c,0f,d8,45,f9,35,a0
1280 g3 DATA 39,27,69,50,ac,02,31,e4,cf
1281 rB DATA 87,3c,00,af,40,0e,51,08,00
1282 12 DATA 5b,7d,5d,e0,59,d9,23,0f,c0
1283 ak DATA 53,35,f8,01,98,00,79,00,a6
1284 10 DATA 8a,9f,12,66,09,c4,04,37,90
1285 yb DATA cb,42,32,34,9c,bb,29,7c,bc
1286 92 DATA 27,76,45,4c,4c,19,6a,c3,36
1287 yg DATA cc,f3,89,dd,ee,4c,ae,50,02
1288 42 DATA b5,e1,1c,3d,b1,60,db,8c,42
1289 Fr DATA 29,48,f0,14,36,24,f8,7e,b8
1290 33 DATA 02,28,00,0a,97,27,db,9b,09
1291 G1 DATA b0,96,4a,c0,37,6c,33,c4,28
1292 QA DATA 43,71,0d,78,4c,ac,39,a8,75
1293 QP DATA fb,81,c7,64,d9,04,dc,ae,a0
1294 b2 DATA b2,ed,2c,9b,03,d7,5b,35,80
1295 Ix DATA 30,b7,c0,a9,d9,1c,0a,44,07
1296 2I DATA 9b,d0,ca,64,07,05,b4,5c,a9
1297 B1 DATA 07,37,e0,bc,dd,fc,2b,2d,0c
1298 fc DATA ad,70,a4,ce,06,00,03,ee,3c
1299 Fz DATA 55,6a,b8,67,54,33,3f,ed,38
1300 W6 DATA ae,01,2f,a8,0c,02,14,a7,04
1301 1W DATA b2,a8,04,48,09,8e,18,c0,41
1302 4o DATA ef,02,c0,82,0d,fa,a3,37,00
```

```
1303 Jb  DATA 01,83,52,37,10,c8,0c,04,01
1304 vb  DATA 02,38,ed,a0,ff,00,0d,7d,ae
1305 Cp  DATA 4c,2b,06,26,20,82,fa,02,ff
1306 8H  DATA 4c,99,05,04,90,05,20,53,3b
1307 e5  DATA 10,c8,8a,12,a0,3c,60,3f,00
1308 KL  DATA 04,d0,26,32,ff,4d,02,40,01
1309 Xz  DATA 17,ff,f9,60,3e,9d,45,08,00
1310 2H  DATA 01,74,6f,70,61,7a,2e,66,6f
1311 eS  DATA 6e,74,ed,6f,38,a8,00,09,02
1312 Zz  DATA 64,7e,3d,df,01,a5,7a,40,52
1313 Wv  DATA 9e,dc,6c,c7,03,98,49,78,cb
1314 Re  DATA 76,da,1a,83,82,81,25,c9,92
1315 4K  DATA 82,75,aa,15,77,74,30,53,bb
1316 p4  DATA 80,56,56,1a,53,77,66,11,48
1317 tv  DATA dc,94,70,b6,4e,63,68,ec,3c'
1318 gJ  DATA e3,96,61,26,1a,33,23,43,6f
1319 HW  DATA ce,30,24,60,9c,8b,27,72,75
1320 Sw  DATA 71,05,0b,48,76,14,eb,4d,61
1321 1r  DATA 6b,65,ee,16,2e,b7,42,30,8e
1322 8h  DATA a8,2b,90,19,77,ac,65,1b,61
1323 Ak  DATA 44,4b,a8,ed,f2,62,75,c8,39
1324 cb  DATA b4,ae,69,96,70,37,61,62,28
1325 o1  DATA a6,9f,a5,0f,37,62,69,20,9e
1326 7s  DATA 85,33,1c,6d,0c,7d,60,1d,08
1327 aK  DATA cc,52,d2,c0,75,ba,96,b3,88
1328 CJ  DATA 55,b9,7a,53,74,e4,0a,b8,78
1329 Rs  DATA 4a,56,65,ca,c0,83,33,6d,34
1330 uH  DATA dc,e5,91,84,2c,5a,72,89,38
1331 T4  DATA 43,61,75,db,29,c2,2e,37,6d
1332 cJ  DATA a7,71,2e,74,4e,e5,a8,cd,b8
1333 10  DATA 1d,1b,56,80,17,73,3e,bf,b4
1334 Dx  DATA 00,32,e6,14,76,dc,a0,ab,18
1335 1z  DATA 27,0b,87,14,0e,cc,68,31,07
1336 LX  DATA 74,f9,02,cb,c0,f7,ec,c8,fa
1337 dr  DATA 93,4b,6a,bb,1b,d1,ba,b9,a4
1338 1d  DATA 6f,76,5c,12,e7,92,48,65,61
1339 1S  DATA 64,ba,01,00,69,de,28,45,6e
1340 18  DATA 71,78,03,75,18,10,e3,00,4e
1341 BB  DATA b0,99,b4,d2,e4,86,bb,36,2a
1342 j4  DATA ea,36,b5,23,f3,80,74,54,61
1343 Ov  DATA 73,6b,71,fe,72,69,1e,1b,76
1344 14  DATA 57,45,78,4b,86,15,00,b9,81
1345 T9  DATA 61,eb,76,d5,dc,82,46,65,c4
1346 DO  DATA 1c,95,53,69,67,5b,55,66,16
1347 3f  DATA 6c,36,63,49,62,46,2e,d1,65
1348 HX  DATA 54,f8,43,70,b6,c8,0d,77,31
1349 kz  DATA 94,40,a7,31,ed,32,dc,dd,d3
1350 Ro  DATA 31,c7,80,70,6c,79,4d,73,f9
1351 cF  DATA 14,b9,41,62,44,46,69,9d,9d
1352 gR  DATA 04,32,d6,95,d0,70,57,42,07
1353 4m  DATA 4f,6c,64,f6,b6,97,2e,2b,02
1354 zf  DATA f7,cc,49,74,46,75,6e,63,88
1355 CY  DATA 07,97,99,81,75,6d,b8,9a,0e
1356 Ae  DATA 09,1b,ee,92,a8,a8,3b,d2,8c
1357 Kb  DATA 39,6c,6f,11,5e,44,65,76,69
1358 v7  DATA 9b,97,44,d0,10,39,e6,0c,53
1359 V4  DATA 5c,85,05,d5,6d,88,cd,6b,24
1360 Xh  DATA e6,57,61,fc,c3,98,53,62,90
1361 w7  DATA 0a,16,8b,1c,41,64,64,dd,c3
1362 4n  DATA 5d,6d,80,45,dd,91,52,65,73
1363 EJ  DATA 6f,76,08,59,90,a4,49,4f,dc
1364 JO  DATA 69,25,bf,2c,4d,61,79,69,6b
1365 1x  DATA 68,2e,99,93,50,75,f2,68,84
1366 86  DATA 06,82,52,61,77,44,6f,46,6d
1367 gA  DATA e0,bf,c0,47,ed,c2,43,43,98
1368 5N  DATA 04,ce,54,79,39,4f,66,4d,65
1369 bR  DATA 6d,84,4f,50,72,6f,63,65,64
1370 XT  DATA 75,ea,47,8c,56,61,63,d1,26
1371 Yp  DATA 88,05,00,e3,4f,52,6e,cd,56
1372 1F  DATA 72,7a,79,2f,97,2f,bc,6b,00
1373 pe  DATA 3c,29,01,da,f0,e8,64,10;1e
1374 wg  DATA 83,cc,c0,c3,2a,f0,76,e1,4f
1375 rt  DATA 55,54,bd,77,61,5d,a3,2c,c0
1376 XC  DATA 2a,f0,80,61,2d,44,4d,41,20
1377 0G  DATA 62,65,f6,87,64,65,c5,3c,f1
1378 Up  DATA 4e,8b,c0,cb,96,77,18,80,97
1379 op  DATA 10,13,f0,b4,08,bc,6c,a1,84
1380 0E  DATA 41,d9,39,78,83,a7,43,6f,70
1381 QK  DATA 70,e2,36,f4,0b,07,f0,40,38
1382 oT  DATA f0,54,76,84,c0,6b,7a,c8,02
1383 67  DATA 06,09,61,73,74,6c,fc,63,6b
1384 7y  DATA 65,c0,8f,55,e0,05,42,6c,69
1385 Hq  DATA 74,ca,b7,20,0a,4c,ca,67,32
1386 qK  DATA 0c,06,e0,1d,37,30,78,83,21
1387 b4  DATA 31,bc,c1,10,32,5e,aa,03,34
1388 BE  DATA 20,41,75,64,cf,8f,64,48,0f
1389 CV  DATA 89,20,4b,15,03,61,c9,33,18
1390 UG  DATA 63,e1,b0,0a,73,bc,0d,2e,fe
1391 r0  DATA 49,4e,2d,70,75,66,66,68,ff
1392 oC  DATA 8a,76,6f,6c,6c,5e,22,81,35
1393 kE  DATA 20,44,ef,6b,53,79,6e,63,68
1394 61  DATA 72,04,bf,69,73,61,74,c0,0d
1395 Xg  DATA 73,77,c8,0c,74,62,15,43,49
1396 jD  DATA 41,2d,42,2c,f8,de,70,61,6e
1397 eg  DATA 73,69,6f,6e,50,6f,72,a9,7e
1398 S1  DATA 16,0d,ab,c0,69,cb,6e,65,37
1399 6W  DATA 20,53,6f,66,74,77,61,72,65
1400 75  DATA f2,73,52,51,70,00,45,78,65
1401 Fg  DATA 63,4c,69,62,2e,f1,1f,2d,36
1402 Iw  DATA 20,4e,4d,49,2d,49,6e,74,65
1403 8U  DATA 72,72,75,70,74,f7,00,fe,03
1404 ct  DATA 1b,20,20,20,d0,06,da,f0,80
1405 3a  DATA 16,72,bc,06,3a,66,69,6c,65
1406 oa  DATA 6e,61,6d,65,f9,35,38,f0,04
1407 mb  DATA 03,b0,d6,40,d8,18,00,b1,0b
1408 T2  DATA bc,a0,58,02,48,00,c5,3c,b1
1409 WE  DATA ed,c1,09,08,7a,65,2d,04,4f
1410 Sh  DATA 69,c0,23,5d,68,32,0b,67,54
1411 w1  DATA 30,70,80,a7,cd,51,71,bc,b5
1412 41  DATA 04,0f,9a,36,e0,ed,e1,03,bc
1413 VC  DATA 9a,38,67,81,80,97,85,97,f0
1414 fj  DATA 30,60,05,1e,16,6e,c2,a3,a4
1415 Bs  DATA c0,5b,bf,25,85,2b,19,7f,fe
1416 Fm  DATA 97,bf,6f,f0,3f,f9,0c,c6,fd
1417 WJ  DATA e2,2c,84,32,68,45,66,d9,10
1418 qT  DATA 80,5e,04,fc,49,32,33,ce,18
1419 yM  DATA a7,01,b7,93,2b,41,03,18,2f
1420 f3  DATA 02,07,fe,5e,0a,02,bc,6e,c0
1421 0g  DATA 99,38,0c,14,83,f8,78,5e,b0
1422 3D  DATA c0,d3,95,05,81,8d,c0,0d,04
1423 dH  DATA fb,eb,7a,4b,00,06,6f,82,6e
1424 gZ  DATA 27,70,93,80,c0,23,c0,1a,34
1425 hf  DATA f0,80,f6,88,90,c0,ef,eb,0d
1426 YW  DATA 03,f0,22,e2,9c,07,5c,08,0c
1427 oF  DATA 02,bc,16,b1,b6,70,c0,5c,65
1428 Q0  DATA e7,28,18,f0,06,8f,26,04,05
1429 mX  DATA c0,5e,0e,3a,44,5c,f9,57,c3
1430 gT  DATA 16,03,05,1e,1b,dc,ff,9f,4c
1431 1F  DATA 7c,6b,ca,00,0f,10,03,60,c1
1432 tZ  DATA 53,87,59,15,ac,b3,00,c0,43
1433 X1  DATA 82,99,0f,11,3e,36,61,05,a8
1434 PE  DATA e0,31,d7,ec,5b,ee,c1,01,af
1435 co  DATA 45,b0,1e,3c,24,fe,bb,09,ab
1436 6F  DATA 41,02,6f,08,9e,e0,61,d0,59
1437 eE  DATA 50,b2,f0,bc,60,a6,01,e7,37
1438 HZ  DATA f1,40,0a,02,e0,bc,01,e4,e3
1439 db  DATA 94,56,80,0a,5e,72,9c,5b,b0
1440 iu  DATA c0,e1,4b,15,e1,3d,98,30,c0
1441 H1  DATA b3,a7,d5,fb,41,2b,c8,0a,c2
1442 G8  DATA 1b,b2,07,07,3c,a6,b6,1b,bc
1443 Wp  DATA 27,61,a0,c0,8b,31,60,82,c7
1444 ZP  DATA 20,04,b2,c4,e0,dd,91,0a,ca
1445 Yf  DATA 90,6b,84,9b,c1,88,df,0b,0f
1446 1D  DATA 5e,68,7f,ff,bf,b7,0b,0d,1f
1447 yd  DATA 68,fe,ff,df,b4,03,3f,74,82
1448 Rp  DATA 19,81,8a,25,c7,93,88,08,48
1449 Ni  DATA f0,18,84,81,53,e8,b7,2a,6c
1450 cM  DATA 03,ff,81,fc,9d,40,c0,fe,3f
1451 pl  DATA 0d,80,03,7f,80,b3,c0,8f,09
1452 3s  DATA 1a,78,08,7e,01,fc,f4,7e,03
1453 Yy  DATA fc,01,68,62,5a,93,b4,ff,01
1454 1W  DATA f8,7c,07,c0,fc,3e,01,fe,fa
1455 nn  DATA b7,6e,00,6d,96,cf,80,3f,ed
1456 Az  DATA 67,7a,80,e7,c4,e2,01,4d,78
1457 zg  DATA 40,0b,1e,d0,80,97,16,97,f0
1458 sk  DATA ce,e2,1d,50,6c,c0,03,80,2d
1459 6G  DATA 79,cf,4b,1e,d0,88,07,34,e1
1460 uc  DATA 05,a9,e0,7d,0f,78,10,0c,78
1461 vT  DATA 50,03,1e,00,03,3c,5e,2f,41
1462 VA  DATA ea,03,cf,dc,1c,54,40,41,ea
1463 v3  DATA 01,ce,0c,34,e0,39,b0,c2,55
1464 LS  DATA 0d,e0,0c,c0,cf,7b,a8,1e,e0
1465 2G  DATA fc,c0,39,40,10,6e,86,1c,50
1466 Rb  DATA 80,c0,37,9b,ee,e0,f0,8d,0f
1467 Cs  DATA 6a,87,1c,99,e0,d3,ec,c2,00
1468 CN  DATA af,4a,24,48,35,94,3e,83,11
1469 7H  DATA c0,1f,87,2d,8f,20,c3,64,80
1470 RK  DATA 0d,fc,1f,21,2f,78,e0,91,13
1471 kj  DATA e0,09,08,d9,dc,c4,8d,00,37
1472 kA  DATA 80,ac,57,b2,8f,82,60,82,b3
1473 hO  DATA 3e,34,61,c2,96,75,15,0a,78
1474 dK  DATA 13,36,ec,01,7e,68,f4,e4,00
1475 y4  DATA 3c,5f,c0,e7,48,ac,0f,c0,7f
1476 rn  DATA f0,dd,21,94,23,21,0f,34,1b
1477 ex  DATA c2,a3,03,f0,58,16,9f,cf,04
1478 su  DATA c1,9e,65,7d,00,68,19,38,f0
1479 kq  DATA 83,62,10,e0,ad,c5,fa,d0,1a
1480 Xs  DATA e4,dc,85,01,7e,3f,9e,7b,85
1481 Pl  DATA 45,96,00,0c,1e,83,6d,b8,fd
1482 up  DATA 65,8e,f0,ff,99,05,ef,df,c4
1483 vt  DATA c5,72,18,33,10,7f,6c,af,04
1484 xU  DATA 89,00,0c,9e,86,44,c3,39,91
1485 h8  DATA 58,a2,fe,f4,43,b3,1d,d6,09
1486 ov  DATA 7f,7e,07,61,47,48,e0,91,c3
1487 21  DATA 80,60,7e,b0,c0,73,8f,c7,e1
1488 AP  DATA e5,08,09,7f,bc,17,06,78,06
1489 9b  DATA 39,52,60,16,08,7e,78,43,c7
1490 DB  DATA 68,04,ac,82,8f,c0,58,7e,12
1491 V7  DATA 0f,b8,ca,f1,80,7e,2b,20,0c
1492 X8  DATA 3a,e4,c6,6b,0d,ff,95,58,63
1493 oM  DATA 60,c0,83,9b,47,0d,12,d6,a0
1494 4S  DATA 03,3f,fe,9e,0f,60,a5,c1,11
1495 Gs  DATA 3c,65,98,61,7f,60,13,24,f0
1496 LC  DATA 4c,e8,01,fb,b0,0b,0c,c1,bb
1497 WA  DATA d0,87,ea,3f,f8,60,80,87,57
1498 7L  DATA 02,00,f3,6f,b7,82,c7,57,87
1499 ws  DATA ff,84,2c,03,93,82,c7,ae,95
1500 T4  DATA 83,77,01,ff,6b,06,03,75,1f
1501 gX  DATA c1,e0,7f,19,a7,ff,e3,f0,fe
1502 bE  DATA b4,63,fc,60,bd,14,3f,bc,19
1503 PZ  DATA 8f,e7,f1,6c,26,0f,e3,f1,fc
1504 D4  DATA bb,7f,8e,19,8f,e3,f8,ec,04
1505 tM  DATA f2,90,41,d5,e2,a5,e0,3d,f3
1506 aP  DATA f9,70,83,f7,f1,fe,7e,3f,07
1507 Fn  DATA be,4d,74,cf,e3,fc,fc,dc,02
1508 VD  DATA f8,ae,0e,1d,5a,5e,0a,f9,de
1509 iB  DATA 0c,0f,ff,f2,80,54,c8,a5,a0
1510 tz  DATA a0,fb,5b,42,00,04,6f,8c,c7
1511 a6  DATA e3,9c,fc,41,33,68,40,fc,3f
1512 Ow  DATA bf,8d,81,a5,f8,88,81,f8,cd
1513 tY  DATA 66,f0,e1,32,07,ff,9b,5d,f8
1514 8g  DATA 24,05,f0,dd,6c,06,ef,84,9f
1515 1U  DATA 36,19,01,14,da,7f,f0,62,c0
1516 8o  DATA 16,75,c7,04,e0,30,83,0f,f0
1517 10  DATA ac,f8,23,02,05,03,5e,2d,19
1518 10  DATA 0f,77,9f,c2,7f,80,3f,87,b6
1519 0s  DATA ff,00,1f,e0,3f,0f,80,f8,1f
1520 OV  DATA 87,c0,3f,c0,ff,f8,00,1a,c0
1521 hu  DATA ff,01,f9,f0,07,01,af,93,59
1522 wp  DATA 52,7f,b2,80,4a,bc,e1,08,0f
1523 Ga  DATA e8,c0,fb,1e,f0,e0,62,01,1e
1524 f8  DATA 93,09,c0,ff,80,b9,28,4e,e0
1525 2g  DATA 1d,c5,0d,68,d2,03,1a,f2,80
1526 ln  DATA 46,3c,a0,05,0f,68,c0,03,1a
1527 ga  DATA f0,c3,12,d5,10,54,a0,ff,ff
1528 oH  DATA ff,f0,4e,d4,21,08,c0,2a,47
1529 ce  DATA 11,00,23,51,5f,e6,ac,e1,ec
1530 61  DATA 06,3b,55,19,2e,53,82,6a,00
1531 bP  DATA a2,25,00,4e,e4,a0,b2,00,68
1532 bU  DATA 09,00,ad,00,9e,24,07,d5,00
1533 nr  DATA be,46,0e,2a,8e,00,e6,1c,ea
1534 L8  DATA d4,4a,fe,1a,04,d4,a0,08,06
1535 NS  DATA 1c,35,22,a8,41,26,0d,aa,81
1536 QJ  DATA 01,40,1c,44,d4,a0,60,06,d5
1537 6E  DATA 01,78,44,0e,7c,6a,b4,50,83
1538 Wt  DATA 2a,01,a1,45,06,30,e4,a0,36
1539 R4  DATA 06,3a,35,46,a8,41,54,0d,5a
1540 Yg  DATA 6a,5e,50,83,66,1a,6e,d4,a0
1541 JN  DATA 7e,06,84,35,88,a8,41,55,06
1542 fn  DATA 9c,90,83,a6,1a,ae,d4,a0,cc
1543 Qw  DATA 06,95,06,e0,40,16,06,f8,e0
1544 8C  DATA a0,02,06,0c,35,72,a8,41,8c
1545 Gg  DATA 0d,96,aa,07,f6,00,07,92,35
1546 N7  DATA ba,a8,02,08,d4,2c,0a,4a,c0
1547 DR  DATA 40,19,a5,a0,eb,a8,02,00,01
1548 Sq  DATA 00,00,03,f2,f6,90,80,00,00
1549 qM  DATA 00,03,eb,00,00,1a,3d,00,00
1550 Na  DATA 03,f2
```


Listing. »VirusControl« macht Schluß mit Viren aller Art (Schluß)

Geräteinspektion

Wollten Sie auch schon in einem Programm testen, ob ein externes Laufwerk oder eine Festplatte angeschlossen sind? Ein Kinderspiel mit »TestDev.h«.

Immer, wenn ein Programm zum Speichern auf ein anderes als das interne Laufwerk zugreifen möchte, sollte getestet werden, ob das entsprechende Laufwerk tatsächlich vorhanden ist. Bei den maximal vier Diskettenlaufwerken stellt dies ja noch kein Problem dar:

```
#include <devices/trackdisk.h>
struct IOExtTD testReq;
LONG DRIVE1 = GADGDISABLED;
if(!(OpenDevice("trackdisk.device",1L,&testReq,0L)
))
{
   CloseDevice(&testReq);
   DRIVE1 = 0L;
}
```

In diesem Beispiel wird für eine spätere Gadget-Abfrage das Flag für die Darstellung als »Ghost«-Gadget gesetzt und, wenn das Laufwerk vorhanden ist, wieder gelöscht (0L in der letzten Zeile). Für die vier Laufwerke müssen in der IF-Abfrage Werte zwischen 0L und 3L eingesetzt werden.

Alles schön und gut, aber eine Festplatte oder das RAM kann so nicht getestet werden, da sie nicht über das trackdisk.device angesprochen werden. Da aber jedes angeschlossene und gemountete Device beim Start des Amiga in eine Liste eingetragen wird, ist es am einfachsten, in dieser Liste nach dem entsprechenden Device zu suchen.

Die Amiga-Entwickler haben es aber nicht leicht gemacht, die relevanten Daten zu finden. Ein Programm muß sich zuerst durch mehrere Systemlisten hangeln. Der Weg führt über DOSBase, RootNode und DosInfo hin zur DeviceList. Deren Definitionen kann der interessierte Programmierer der Include-Datei »libraries/dosextens.h« entnehmen.

Dort angekommen, kann die in der Include-Datei »libraries/filehandler.h« definierte Struktur ausgelesen werden. Hierbei offenbart sich ein weiteres Hindernis. Einige der Zeiger sind BCPL-Zeiger (BPTR), die erst umgerechnet werden müssen. Dazu dient die Definition in Zeile 15 in Listing 1, die das BCPL-Format durch Multiplikation mit dem Wert 4 in das übliche C-Adreßformat umrechnet. In der while-Schleife wird das Element »dn_Type« durchsucht und mit der Konstanten »DLT_DEVICE« verglichen. Nur wenn der Vergleich positiv ist, wird der Eintrag weiterbearbeitet. Da hier noch andere Knotentypen stehen, wir aber nur nach echten Devices suchen, werden die anderen Typen so übergangen.

In den Zeilen 25 bis 29 wird der so gefundene Name in einen C-String umgewandelt. Hierzu eine kurze Erläuterung: Bei einem BCPL-String gibt das erste Byte die Länge des Strings an, die folgenden Bytes sind die einzelnen Zeichen. Ein C-String dagegen wird durch ein Nullbyte abgeschlossen. Der so gewonnene String wird in Zeile 33 mit dem beim Aufruf der Unterroutine übergebenen Ausdruck verglichen. Stimmt der Vergleich, kehrt die Unterroutine mit dem Ergebnis TRUE zurück. Beim Aufruf von TestDev muß das zu suchende Device ohne Doppelpunkt übergeben werden. Wie das Ergebnis, das TestDev liefert, später vom Hauptprogramm ausgewertet wird, hängt ganz allein von Ihnen ab. Auf alle Fälle verfügen Sie nun über eine Kontrollmöglichkeit, die das Vorhandensein eines Devices anzeigt.

Dieter Mayer/kn

Programmname:	TestDev.h
Bemerkung:	Nur zum Einbinden

```
Programmautor: Dieter Mayer
---------------------------
 1 hK0 /* Include-Datei "TestDev.h" Testet, ob
 2 Y5  vorgegebenes Device angeschlossen ist. */
 3 bN  #ifndef LIBRARIES_FILEHANDLER_H
 4 Hf3    #include <libraries/filehandler.h>
 5 MI0 #endif
 6 8h  #ifndef LIBRARIES_DOSEXTENS_H
 7 Oy3    #include <libraries/dosextens.h>
 8 PL0 #endif
 9 ug  TestDev(device)
10 vD  char *device;
11 7a  {
12 SQ3    char DeviceName[10], *zeiger;
13 hm     BOOL erg = FALSE;
14 90     int zahl;
15 Wg     #define GetPtr(x) ((APTR)(((ULONG)(x))<<2))
16 bV     struct DeviceNode *devinfo, *next;
17 Kc     extern struct DosLibrary *DOSBase;
18 8R     struct DosInfo *info;
19 dV     struct RootNode *root;
20 70     root = DOSBase->dl_Root;
21 uD     info = GetPtr(root->rn_Info);
22 FY     devinfo = GetPtr(info->di_DevInfo);
23 uF     while(devinfo)
24 Kn     {
25 BC6       next=GetPtr(devinfo->dn_Next);
26 Ce        if(devinfo->dn_Type==DLT_DEVICE)
27 Nq        {
28 BN9          zeiger=GetPtr((BPTR)devinfo->dn_Name);
29 Om           zahl=*zeiger++;
30 Jb           if(zahl>10) zahl=10;
31 QJ           strncpy(DeviceName,zeiger,zahl);
32 f9           DeviceName[zahl]=0;
33 Ag           if(strcmp(DeviceName,device)==0) erg=TRUE;
34 Y36       }
35 eH        devinfo=next;
36 a53    }
37 7X     return erg;
38 c70 }
(C) 1989 M&T
```

Listing 2. »TestDev.h« testet das Vorhandensein von Devices (Geräten).

Programmname:	Demo.c
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.4, V3.6
Aufrufe:	CC Demo +L -S LN Demo.o -LC32

```
Programmautor: Dieter Mayer
---------------------------
 1 z00 /*
 2 DW  ** Demoprogramm zum Testen, ob ein bestimmtes
 3 07  ** Device angeschlossen ist.
 4 sC  **
 5 VF  **          12/88 by Dieter Mayer
 6 9Y  */
 7 8K  #include "TestDev.h"
 8 IP  UBYTE Entry[][5]=
 9 5Y  {
10 dQ3    (UBYTE *)"DF0",(UBYTE *)"DF1",(UBYTE *)"DF2",
11 IM     (UBYTE *)"DF3",(UBYTE *)"JH0",(UBYTE *)"JH1",
12 xq     (UBYTE *)"DH0",(UBYTE *)"DH1",(UBYTE *)"DH2",
13 Ya     (UBYTE *)"DH3",(UBYTE *)"DH4",(UBYTE *)"DH5",
14 1s     (UBYTE *)"RAD",(UBYTE *)"VD0",(UBYTE *)"VDK",
15 9x     (UBYTE *)"CON",(UBYTE *)"RAM",(UBYTE *)"AUX",
16 RR     (UBYTE *)"CARD",(UBYTE *)"PRT",(UBYTE *)"SER",
17 MB     (UBYTE *)"FAST",(UBYTE *)"PAR",(UBYTE *)"RAW"
18 bX0 };
19 OM  main()
20 Gj  {
21 pL3    int i;
22 G3     BOOL erg;
23 X3     for(i=0;i<24;i++)
24 Kn     {
25 u16       erg=TestDev(Entry[i]);
26 Js        if(erg==TRUE) printf("Mounted device: %s\n",Entry[i]);
27 Rw3    }
28 Sx0 }
(C) 1989 M&T
```

Listing 1. »Demo« zeigt die Anwendung der Header-Datei »TestDev.h«

Dämmerung

Schöne Effekte versüßen das Arbeiten und Spielen am Computer. Das Ein- und Ausblenden von Bildern gehört dazu. »Fade« erledigt das für Sie.

Öffnet ein Programm einen neuen Bildschirm (Screen), erscheint dieser schlagartig auf dem Monitor. Manchmal wünscht man sich aber ein weiches Einblenden. »Fade« besteht aus zwei C-Routinen, die genau das bewirken.

Zunächst zum Ausblenden. Die Routine heißt »FadeOut«. Sie verlangt folgende Parameter:
— sc: Ein Zeiger auf den auszublendenden Bildschirm,
— Time: Die Verzögerung beim Vorgang in 1/50 Sekunden.

Je größer Time ist, um so länger dauert das Ausblenden.

Das Gegenstück zu »FadeOut« ist »FadeIn«. Es verlangt drei Übergabeparameter:

Aufruf Lattice-C-Compiler:	LC -L Fade.c
Aufruf Aztec-C-Compiler:	CC Fade +L
	LN Fade.o -LC32

— sc und Time: wie bei FadeOut
— Colors: Zeiger auf eine Farbtabelle

»sc« und »Time« haben dieselbe Bedeutung wie bei »FadeOut« beschrieben. Die Farbwerte in »Colors« geben die zu erreichenden Werte an. Dafür muß für jede Farbe ein Wert in einem Feld von UWORD-Variablen vorhanden sein. *A. Schwarzhaupt/R. Beaupoil/kn*

Programmname:	Fade
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Bemerkung:	siehe Text

```
Programmautor: Andreas Schwarzhaupt
-----------------------------------

1GO  /* Demo.c : Demonstriert die Funktionen FadeIn und FadeOut
n14      Übersetzbar mit Aztec- und Lattice-C-Compiler
kA       written in 1989 by Andreas Schwarzhaupt */
L90  #include <intuition/intuitionbase.h>
Y1   struct IntuitionBase *IntuitionBase;
sV   struct GfxBase *GfxBase;
in   /* Fadeout () : Fadeout eines Screens
6S3     Parameter  : Zeiger auf Screen - Struktur
YPG                  Dauer des Fades (2) */
1t0  void FadeOut (sc, Time)
7q   struct Screen *sc;
kH   BYTE Time;
9c   {
HS2    BOOL AllBlack;
Ts     BYTE Col;
su     WORD Color, r, g, b, i;
3o     Col = sc->ViewPort.ColorMap->Count;
yf     AllBlack = FALSE;
AX     while (!AllBlack)
Gj     {
XB8          AllBlack = TRUE;
60           for (i = 0; i < Col; i++)
Jm           {
leC              if (Color = GetRGB4 (sc->ViewPort.ColorMap, i)
                 )
5mF                 AllBlack = FALSE;
22C              r = Color>>8;
f6               g = (Color-(r<<8))>>4;
9f               b = Color-(r<<8)-(g<<4);
ub               if (((r >= g) && (r >= b)) && (r > 0))
mxF                 r--;
WrC              if (((g >= r) && (g >= b)) && (g > 0))
SSF                 g--;
QbC              if (((b >= g) && (b >= r)) && (b > 0))
KFF                 b--;
QkC              SetRGB4 (&sc->ViewPort, i, r, g, b);
a58          }
LR           Delay (Time);
c72    }
d80  }
WJ   /* FadeIn () : FadeIn eines Screens
j23     Parameter : Zeiger auf Screen,
9EF                 Dauer des Fades,
1c                  Zeiger auf Farb - Endwert - Tabelle */
Ue0  void FadeIn (sc, Time, Colors)
f0   struct Screen *sc;
Ip   BYTE Time;
Ce   UWORD *Colors;
1B   {
go2    BYTE r, g, b, rv, gv, bv, Col;
8G     WORD i, Color;
9y     BOOL AllIn;
cN     Col = sc->ViewPort.ColorMap->Count;
Tb     AllIn = FALSE;
Lv     while (!AllIn)
pI     {
5f8          AllIn = TRUE;
fZ           for (i = 0; i < Col; i++)
sL           {
DqC              if (!((Color = GetRGB4 (sc->ViewPort.ColorMap,
                  i)) == Colors [i]))
aiF                 AllIn = FALSE;
n2C              r = Color >> 8;
Ij               g = (Color-(r<<8)) >> 4;
iE               b = Color-(r<<8)-(g<<4);
Co               rv = Colors [i] >> 8;
Di               gv = (Colors [i]-(rv<<8)) >>4;
OX               bv = Colors [i]-(rv<<8)-(gv<<4);
zV               if (r < rv)
AHF                 r++;
47C              if (g < gv)
qmF                 g++;
DVC              if (b < bv)
iZF                 b++;
2MC              SetRGB4 (&sc->ViewPort, i, r, g, b);
Ch8          }
x3           Delay (Time);
EJ2    }
Fk0  }
gP   int main (argc, argv)
hH   int argc;
Wd   char *argv [];
Fi   {
xU2    struct Screen *Scr;
GQ     int i, col;
NQ     UWORD Colors [32];
rR     if(!(GfxBase=(struct Library *)OpenLibrary("graphics.libr
       ary",0L)))
Hy     exit(30);
Xq     if(!(IntuitionBase=(struct Library *)OpenLibrary("intuiti
       on.library",0L)))
J0     exit(30);
VV     Scr=IntuitionBase->ActiveScreen;
TI     col = Scr->ViewPort.ColorMap->Count;
1B     for (i = 0; i < col; i++)
Qt     {
KY6        Colors [i] = GetRGB4 (Scr->ViewPort.ColorMap, i);
Wg         SetRGB4 (&Scr->ViewPort, i, 0, 0, 0);
X22    }
8i     ScreenToFront (Scr);
Tr     FadeIn (Scr, 4, Colors);
5V     FadeOut (Scr, 2);
Fa     FadeIn (Scr, 10, Colors);
2Y     CloseLibrary(GfxBase);
uT     CloseLibrary(IntuitionBase);
e90  }
(C) 1989 M&T
```

Listing. Die zwei Routinen »FadeIn« und FadeOut« mit Demoteil. Bitte mit dem Checksummer (Seite 143) eingeben.

BITTE UMBL

Ein großes Plus des Amiga sind die Fenster. Viele Informationen lassen sich so gleichzeitig darstellen. Doch wie kommt man schnell und einfach an das hinterste Fenster? »Shuffle« gibt Ihnen die Antwort.

Ein Programm nimmt Ihnen die Arbeit ab, Fenster nach hinten zu klicken oder zu schließen. Außerdem steckt in dem Listing »Shuffle« hochinteressanter Stoff: Es handelt sich nämlich um einen Input-Handler. »Shuffle« (mischen) hängt sich hinter das Input-Device und vor die anderen Programme. Drückt man die Taste <Help>, legt es das nächste Fenster nach oben, über dem der Mauszeiger steht. Sie müssen also nicht alle Fenster »durchblättern«, bis Sie zum gewünschten gelangen. Außer <Help> erkennt Shuffle noch die Tastenkombination <rechte Amiga Esc>. Damit schalten Sie das Programm aus. Es wird komplett aus dem System herausgenommen.

Doch wie macht man so etwas? Anhand des Listings erläutern wir es. Einige Dinge sind wichtig, wenn man ein Programm wie Shuffle schreiben möchte. Zunächst stellt sich die Frage, wie man einen Hintergrundprozeß startet. Bei Verwendung des Aztec-C-Compilers verwendet man hierfür zusätzlich zu den benötigten C-Bibliotheken noch »ldetach.o«. Damit dies funktioniert, benötigt das Programm noch zusätzliche Informationen. Die Variablen finden Sie in den Zeilen 76 bis 79.

Das Programm läuft also im Hintergrund. Als nächste Aufgabe steht das Auswerten von Tastendrücken an. Zunächst brauchen wir eine Struktur, die die Adresse unseres Tasks sowie die Signale und die entsprechenden Nummern enthält. Diese Struktur heißt »GLOBAL__DATA« und wird in den Zeilen 15 bis 22 definiert. Die verwendete Struktur hat dann den Namen »global«. Das Eintragen der Werte in die Struktur erfolgt im Hauptprogramm in den Zeilen 109, 119, 123, 124 und 128. Um die Task-Adresse zu ermitteln, verwenden wir die Funktion FindTask(0L) (Zeile 109). Für die Belegung der Signale gibt es die Funktion AllocSignal() (Zeilen 119 und 124). Der Parameter -1 besagt, daß wir keine spezielle Tasknummer wünschen. Das Bit wird dann einfach durch Verschieben des Wertes 1 gesetzt (Zeilen 123 und 128). Die Struktur »global« ist jetzt vollständig gefüllt.

Die nächste erforderliche Struktur ist »handlerStuff« (Zeile 105). Es handelt sich dabei um eine Interrupt-Struktur. Diese wird nun in den Zeilen 137 bis 139 gefüllt. Als erstes kommt der Zeiger auf die schon definierte Struktur »global«. Es folgt ein Zeiger auf die Interface-Routine (»HandlerInterface«). Sie ist in Assembler geschrieben, da die Parameter in den Registern A0 und A1 geliefert werden. Damit kann aber die eigentliche Interrupt-Routine nichts anfangen, da sie in C geschrieben ist. Die Registerinhalte werden auf den Stack gelegt und die Funktion »myhandler« aufgerufen. Die letzte der Zeilen setzt die Priorität des Interrupts. In den Zeilen 144 bis 146 binden wir unseren Interrupt-Handler ins System ein.

In der »Endlos«-Schleife erfolgt die Auswertung eines aufgetretenen Signals. Mit der Funktion Wait() legen wir Shuffle schlafen, bis ein Signal empfangen wird. Erhalten wir das Signal zum Beenden, verlassen wir die Schleife und geben die vom System angeforderten Bibliotheken, Signale etc. wieder frei. Danach endet unser Programm. Empfangen wir jedoch das Signal für Aktion, holen wir das nächste Fenster nach vorn.

Die eben erwähnten Signale erhalten wir von der Routine »myhandler« (Zeile 29 bis 71). Hier erfolgt die Auswertung der gedrückten Taste. Handelt es sich um die Tastenkombination <rechte Amiga Esc>, schicken wir das Signal zum Beenden. Ist es die <Help>-Taste, geben wir das Signal zur Aktion. Trifft keiner der beiden Fälle zu, geben wir den Zeiger auf die Event-Liste zurück. Dadurch kann der nächste Handler arbeiten, als wäre nichts passiert.

Natürlich können Sie statt des Blätterns der Fenster auch andere Dinge erledigen. Achten Sie jedoch unbedingt darauf, daß Ihre Routine so wenig Zeit wie irgend möglich benötigt. Das ist wichtig, damit das System nicht unnötig belastet wird. Falls Ihnen die Tastenbelegung von Shuffle nicht gefällt, ändern Sie diese in den Zeilen 12 bis 13.

Shuffle arbeitet mit den Fenstern des aktuellen Bildschirms. Dadurch ist es universell einsetzbar, auch in Programmen, die einen eigenen Screen öffnen.

René Beaupoil/kn

Programmname:	Shuffle
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.6
Aufrufe:	CC Shuffle +C +D +L LN Shuffle.o ldetach.o32 -LCL32

```
Programmautor: Stefan Auerbach
------------------------------

 1 5J0 /*************************/
 2 sF  /* Shuffle Version 2.0    */
 3 xz  /* Autor: Stefan Auerbach */
 4 ZF  /* Philipp-Reis-Str. 1    */
 5 yT  /* 6450 Hanau 1           */
 6 Ao  /*************************/
 7 Oo  /* * * * * * * * * INCLUDE FILES * * * * * * * * * * * */
 8 iM  #include <functions.h>
 9 2q  #include <devices/input.h>
10 RF  #include <intuition/intuitionbase.h>
11 yG  /* * * * * * * * * * * CONSTANTS * * * * * * * * * * * * */
12 aG  #define ESCAPEKEY    0x45      /* Tastencode für Esc-Taste
         */
13 Xm  #define HELPKEY      0x5f      /* Tastencode für Help_Tast
       e */
14 am  /* * * * * * * * * * * GLOBAL VARIABLES * * * * * * * * * *
       /
15 rx  typedef struct
16 Cf  {
17 g83    struct Task          *TaskToSig;    /* Adresse des warte
          nden Tasks  */
18 vC     ULONG                 QuitSig;      /* Signal um Handler
           zubeenden  */
19 8C     ULONG                 QuitSigNr;    /* Signal Nr. für Fr
          eeSignal()  */
20 e0     ULONG                 ActionSig;    /* Signal zum Blätte
          rn          */
21 65     ULONG                 ActionSigNr;  /* Signal Nr. für Fr
          eeSignal()  */
22 xa0 } GLOBAL_DATA;
23 iQ  GLOBAL_DATA          global;
24 dA  struct MsgPort     *inputDevPort;
25 ge  struct IOStdReq    *inputRequestBlock;
26 OK  /*******************************************************
       ******/
27 Ga  /* Die Handlerroutine, die vom Handlerinterface aufgerufen
       wird */
28 2M  /*******************************************************
       ******/
29 1Z  struct InputEvent *myhandler(ev, gdptr)
30 er  struct InputEvent *ev;         /* Pointer auf die Eventliste
           */
31 hL  register GLOBAL_DATA *gdptr;  /* Alle benötigten Informatio
       nen */
32 Sv3    {
33 QD     register struct InputEvent *ep, *laste;
34 13     /* Die Eventliste durchsuchen um zu sehen,
              */
35 Dc     /* ob die richtige Taste  gedrückt wurde
              */
36 x1     for (ep = ev, laste = NULL; ep != NULL; ep = ep->ie_Nex
          tEvent)
37 X06       {
38 Ly        if ((ep->ie_Class == IECLASS_RAWKEY) &&
39 VWA           (ep->ie_Code  == ESCAPEKEY)         &&
40 aI            (ep->ie_Qualifier & IEQUALIFIER_RCOMMAND))
41 b49          {
42 r5           /* Falls wir den Event bearbeiten können =>  */
43 gJ           /* Event aus der Liste löschen.             */
44 r4           if (laste == NULL)
45 NFC             ev = ep->ie_NextEvent;
46 ZM9          else
```

ÄTTERN...

```
            laste->ie_NextEvent = ep->ie_NextEvent;
          /* Signalisiere, daß der Prozess enden soll */
          Signal(gdptr->TaskToSig, gdptr->QuitSig);
          }
        else
          {
          if ((ep->ie_Class == IECLASS_RAWKEY) &&
             (ep->ie_Code  == HELPKEY))
            {
            /* Falls wir den Event bearbeiten können =>  *
            /
            /* Event aus der Liste löschen.              */
            if (laste == NULL)
              ev = ep->ie_NextEvent;
            else
              laste->ie_NextEvent = ep->ie_NextEvent;
            /* Signalisiere, Windowpop */
            Signal(gdptr->TaskToSig, gdptr->ActionSig);
            }
          else
            laste = ep;
          }
      }
    /* Pointer auf die Eventliste an den nächsten Handler we
    iterreichen. */
    return(ev);
    }
/* * * * * * * * * * * EXTERNAL ROUTINES * * * * * * * * *
*/
struct IntuitionBase *IntuitionBase = NULL;
LONG   LayersBase = NULL;
/* für die detach Funktion von Aztec-C V3.6a notwendig */
LONG                   _stack = 10000;
LONG                   _priority = 0;
LONG                   _BackGroundIO = 1;
char                  *_procname = "Shuffle";
HandlerInterface()
{
 #asm
 movem.L A0/A1,-(A7)
 jsr _myhandler
 addq.L #8,A7
 #endasm
}
CleanUp()
{
    if(inputRequestBlock)CloseDevice(inputRequestBlock);
    if(IntuitionBase)CloseLibrary(IntuitionBase);
    if(LayersBase)CloseLibrary(LayersBase);
    if(global.ActionSigNr)FreeSignal(global.ActionSigNr);
    if(global.QuitSigNr)FreeSignal(global.QuitSigNr);
    if(inputRequestBlock)DeleteStdIO(inputRequestBlock);
    if(inputDevPort)DeletePort(inputDevPort);
}
/*********************************/
/* Das main-Programm für Shuffle */
/*********************************/
void main()
{
    LONG  X_Coord = 0, Y_Coord = 0; /* Mauskoordinaten im Sc
    reen */
    ULONG sig;
    struct Interrupt   handlerStuff;
    struct Layer_Info  *li;
    struct Layer       *Layer;
    struct Window      *Window;
    global.TaskToSig = FindTask(0L);
    SetTaskPri( global.TaskToSig, 20L);
    if ((inputDevPort = CreatePort(0L,0L)) == NULL) /* für i
    nput device */
    {
       CleanUp(1);
    }
    if ((inputRequestBlock = CreateStdIO(inputDevPort)) == 0
    )
    {
       CleanUp(2);
    }
    if ((global.QuitSigNr = AllocSignal(-1L)) == -1)
    {
       CleanUp(3);
    }
    global.QuitSig = 1 << global.QuitSigNr;
    if ((global.ActionSigNr = AllocSignal(-1L)) == -1)
    {
       CleanUp(4);
    }
    global.ActionSig = 1 << global.ActionSigNr;
    if ((LayersBase = (LONG) OpenLibrary("layers.library", 0
    L)) == NULL)
    {
       CleanUp(5);
    }
    if ((IntuitionBase = (LONG) OpenLibrary("intuition.libra
    ry", 0L)) == NULL)
    {
       CleanUp(6);
    }
    handlerStuff.is_Data = (APTR)&global;
    handlerStuff.is_Code = HandlerInterface;
    handlerStuff.is_Node.ln_Pri = 52;
    if (OpenDevice("input.device",0L,inputRequestBlock,0L))
    {
       CleanUp(7);
    }
    inputRequestBlock->io_Command = IND_ADDHANDLER;
    inputRequestBlock->io_Data    = (APTR)&handlerStuff;
    DoIO(inputRequestBlock);
    for(;;)          /* FOREVER */
    {
       /* Let's wait auf ein Signal vom Handler */
       sig = Wait( global.QuitSig | global.ActionSig);
       if (sig & global.QuitSig)
          break;
       if (sig & global.ActionSig)
       {
          Disable();
          /* Hole LayerInfo Pointer */
          li = &(IntuitionBase->ActiveScreen->LayerInfo);
          /* Sperre LayerInfo für andere Tasks */
          LockLayerInfo(li);
          /* Hole Mauskoordinaten */
          X_Coord = (LONG)(IntuitionBase->ActiveScreen->Mo
          useX);
          Y_Coord = (LONG)(IntuitionBase->ActiveScreen->Mo
          useY);
          /* Hole Pointer auf die zugehörige Layer-Struktur
          */
          Layer   = (struct Layer *)WhichLayer(li,X_Coord,Y_
          Coord);
          /* Hole Pointer auf die zugehörige Window-Struktur
           */
          Window  = (struct Window *)Layer->Window;
          /* Push oder Pop Window */
          if (Layer == li->top_layer)
             WindowToBack(Window);
          else
          {
             WindowToFront(Window);
        ActivateWindow(Window);
          }
          /* LayerInfo freigeben */
          UnlockLayerInfo(li);
          Enable();
       }
    }
    /* Aufräum Arbeiten */
    inputRequestBlock->io_Command = IND_REMHANDLER;
    inputRequestBlock->io_Data    = (APTR)&handlerStuff;
    DoIO(inputRequestBlock);
    CleanUp(0);
}
(C) 1989 M&T
```

Listing. Verdeckte Fenster? »Shuffle« bringt das gewünschte Fenster schnell und problemlos nach oben. Bitte mit dem Checksummer (Seite 143) eingeben.

Das Arbeiten mit numerischen Feldern in Amiga-Basic verschlingt viel Zeit. Das muß nicht sein. Ein kleiner Umweg macht Ihre Programme wesentlich schneller.

Der schne

von René Beaupoil und Matthias Rustler

Es gibt zwei Wege mit dem Auto vom Münchner Norden in den Süden zu kommen.

— Der kurze Weg quer durch die überfüllte Stadt.

— Der längere Weg über den Autobahnring, auf dem man (außer in der Urlaubszeit) flott vorwärtskommt.

Im zweiten Fall kommen Sie schneller ans Ziel. Die Basic-Subprogramme von »Felder« machen Ihre Programme etwas länger, beschleunigen sie aber erheblich. Der Trick an der Sache ist, daß sich Zeichenketten (Strings) schneller manipulieren lassen als numerische Felder. Doch wie wandelt man diese Felder in Strings um? Das Betriebssystem hilft uns.

Der benutzte Befehl stammt aus der »exec.library« und heißt »CopyMem()«. Er kopiert einen beliebig langen Speicherbereich in einen anderen. Die Subprogramme kopieren also numerische Felder in eine Zeichenkette und nehmen dort die gewünschte Manipulation vor. Anschließend kopieren Sie den String wieder zurück.

Um die Routinen zu verwenden, müssen Sie sie zu Ihrem Basic-Programm hinzufügen. Dies geschieht mit dem Befehl MERGE. Außerdem muß im selben Verzeichnis, in dem Amiga-Basic und das Programm steht, auch die Datei »exec.bmap« vorhanden sein. Diese Datei finden Sie auf Ihrer Extras-Diskette im Verzeichnis »BasicDemos«.

Es stehen Ihnen sechs Routinen zur Verfügung, die nun eingehend besprochen werden.

☐ FastLoad `FastLoad feld%(),"Name",puffer`

Diese Routine lädt Werte aus einer sequentiellen Datei in das angegebene Feld. Der Parameter »feld%()« ist der Name des numerischen Feldes, das Sie benutzen. Mit »Name« gibt man den kompletten Zugriffspfad für die Datei an. »puffer« bestimmt, wie groß der Speicherbereich ist, der zum Laden verwendet wird. Je größer dieser Wert ist, um so schneller wird geladen. Andererseits verbraucht dies auch mehr Speicher.

☐ FastSave `FastSave feld%(),"Name",puffer`

Dieses Kommando ist das Gegenstück zum Befehl »FastLoad«. Die Parameter haben dieselbe Bedeutung.

☐ Init `Init feld%(),anfang,ende,wert`

Hiermit initialisieren Sie das Feld »feld%()«. Normalerweise sind alle Feldelemente 0. Sie können jedoch einem beliebigen Teil einen bestimmten Wert (»wert«) zuweisen. Dies geschieht vom Element

Programmname:	Felder
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2
Bemerkung:	siehe Text

```
Programmautor: Matthias Rustler
-------------------------------

1e0 LIBRARY "exec.library"
sI  CLEAR,30000
gg  DIM p%(1500)
eL  PRINT "Initialisieren mit der neuen Routine"
tk  v=TIMER
Vq  Init p%(),0,1500,300
Xd  vn=TIMER-v
iE  PRINT "Initialisieren normal"
xo  v=TIMER
vQ  FOR i=0 TO 1500:p%(i)=300:NEXT
el  vo=TIMER-v
LN  GOSUB PrintFeld
L7  PRINT "Mit Feld-Routinen zirka "vo/vn" mal schneller"
aD  PRINT
KC  PRINT "Einfügen mit der neuen Routine"
4v  v=TIMER
do  Insert p%(),750,55
io  vn=TIMER-v
pG  PRINT "Einfügen normal"
8z  v=TIMER
JE  FOR i=1500 TO 751 STEP-1:p%(i)=p%(i-1):NEXT
8V  p%(750)=55
qx  vo=TIMER-v
XZ  GOSUB PrintFeld
XJ  PRINT "Mit Feld-Routinen zirka "vo/vn" mal schneller"
mP  PRINT
3V  PRINT "Löschen mit der neuen Routine"
G7  v=TIMER
0E  Del p%(),10,12
u0  vn=TIMER-v
p0  PRINT "Löschen normal"
KB  v=TIMER
aM  FOR i=10 TO 1498:p%(i)=p%(i+2):NEXT
18  vo=TIMER-v
ik  GOSUB PrintFeld
iU  PRINT "Mit Feld-Routinen zirka "vo/vn" mal schneller"
xa  PRINT
7Q  PRINT "Suchen mit der neuen Routine"
RI  v=TIMER
pI  Search p%(),0,55,x%
5B  vn=TIMER-v
9C  PRINT "Suchen normal"
VM  v=TIMER
9o  FOR i=0 TO 1500:IF p%(i)=55 THEN rrr
AQ  NEXT i
k8  rrr:
EL  vo=TIMER-v
Mt  PRINT "55 hat den Index: ";x%
vh  PRINT "Mit Feld-Routinen zirka "vo/vn" mal schneller"
An  PRINT
JW  PRINT "Drücken Sie den linken Mausknopf"
oP  WHILE MOUSE(0)=0:WEND
PV  CLS
VF  ERASE p%
cy  g=6022
Yr  DIM p%(g)
b1  FOR x=0 TO 300 STEP 10
Ym2   LINE (x,0)-(300-x,150),RND*3
270 NEXT
Ea  GET (0,0)-(300,150),p%
72  LOCATE 20,5
Ax  PRINT "Speichern mit der neuen Routine"
pg  v=TIMER
eR  FastSave p%(),"Bild",4000
TZ  vn=TIMER-v
HD  LOCATE 21,5
Fr  PRINT "Speichern normal"
ul  v=TIMER
ph  OPEN "Bild2" FOR OUTPUT AS 1
fP  FOR i=0 TO 6022
op2   PRINT #1,MKI$(p%(i));
br0 NEXT i
zj  CLOSE 1
fm  vo=TIMER-v
VS  LOCATE 22,5
ZU  PRINT "Zirka "vo/vn" mal schneller"
7H  LOCATE 20,40
NV  PRINT "Laden mit der neuen Routine"
5w  v=TIMER
bi  FastLoad p%(),"Bild",4000
jp  vn=TIMER-v
HS  LOCATE 21,40
HY  PRINT "Laden normal"
A1  v=TIMER
zg  OPEN "Bild2" FOR INPUT AS 1
vf  FOR i=0 TO 6022
U02   a$=INPUT$(2,#1)
az    p%(i)=CVI(a$)
s80 NEXT i
G0  CLOSE 1
w3  vo=TIMER-v
Wi  LOCATE 22,40
ql  PRINT "Zirka "vo/vn" mal schneller"
Eu  PUT (320,0),p%
```

lle Umweg

»anfang« bis zum Element »ende«. Wollen Sie keine Werte ändern, lautet der Aufruf: `Init feld%,0,0,0`

☐ Insert `Insert feld%(),position,wert`

Das Einfügen von neuen Werten in das Feld geschieht schnell und einfach. Mit »position« geben Sie den Feldindex an, an dem das neue Element stehen soll. »wert« stellt den einzufügenden Zahlenwert dar. Die nachfolgenden Feldelemente werden nach hinten verschoben.

☐ Del `Del feld%(),anfang,ende`

Löscht beliebig viele Feldelemente. Die dahinterliegenden Elemente werden um »ende-anfang«-Stellen nach vorne verschoben. »anfang« legt das erste zu löschende Feld fest. Mit »ende« bestimmt man das letzte zu löschende Element.

☐ Search `Search feld%(),anfang,wert,ergebnis%`

Oft sucht man nach einem Wert und benötigt den entsprechenden Feldindex. Damit die Routine flexibel ist und nicht nur den ersten Wert findet, legen Sie mit »anfang« den Index fest, ab dem gesucht wird. Der zu suchende Wert wird im Parameter »wert« übergeben. »ergebnis%« liefert zwei Werte als Ergebnis:

— Bei erfolgreicher Suche enthält »ergebnis%« den Index des Feldelements, das den gesuchten Wert enthält.

— Konnte der Wert nicht gefunden werden, liefert »ergebnis%« -1.

In der vorliegenden Version arbeiten die Subprogramme mit Integervariablen (2 Byte). Wenn Sie andere Datentypen verwenden wollen, müssen Sie folgende Änderungen vornehmen:

— Das Typkennzeichen der Variablen muß geändert werden. Das jetzt vorhandene Zeichen »%« muß durch eines der folgenden ersetzt werden.

&: lange Ganzzahl, 4 Byte
!: Fließkommazahl, einfach genau, 4 Byte
#: Fließkommazahl, doppelt genau, 8 Byte

— »VL%« muß die Zahl der Bytes je Feldelement enthalten.

— Die MKI$-Funktionen müssen, dem Typ entsprechend, in MKL$ (lange Ganzzahl), MKS$ (einfach genaue Fließkommazahl) oder MKD$ (doppelt genaue Fließkommazahl) geändert werden.

Da Zeichenketten in Basic maximal 32767 Zeichen groß sein dürfen, sind die Routinen auch auf diese Länge beschränkt. Die maximale Anzahl der Feldelemente ergibt sich aus 32767/VL%. Ausnahmen bilden die Routinen »FastLoad« und »FastSave«. Hier sind bis zu 32767 Feldelemente erlaubt.

Listing 1 zeigt Ihnen den Geschwindigkeitsvorteil. Dies sind natürlich keine fixen Werte, da es von verschiedenen Faktoren abhängt. Aber Faktoren bis zu 20 treten durchaus auf. Die Zeilen 2 bis 107 können Sie durch Ihr eigenes modifiziertes Programm ersetzen.

Einen Nachteil hat das schnelle Autobahnfahren natürlich: Der Benzinverbrauch ist höher. Bei dem Programm ist es analog: Der Speicherbedarf bei Verwendung der Routinen ist höher. Aber so lange Sie genügend Sprit haben... *kn*

```
 95 BP  LIBRARY CLOSE
 96 ID  END
 97 gA  PrintFeld:
 98 mE  PRINT "Index: ";
 99 FN  FOR a=0 TO 15
100 DX2   PRINT USING "####";a;
101 in0 NEXT
102 07  PRINT :PRINT "Wert:  ";
103 JR  FOR a=0 TO 15
104 wW2   PRINT USING "####";p%(a);
105 mr0 NEXT
106 4h  PRINT
107 5h  RETURN
108 6C  SUB FastLoad (f%(),File$,puffl%) STATIC
109 20  VL%=2
110 Bx  IF puffl%<1 THEN puffl%=1024
111 Ja  OPEN File$ FOR INPUT AS 1
112 ep2   lok%=0
113 GD    lges%=LOF(1)
114 r5    IF (UBOUND(f%)+1)*VL% < lges% THEN ERROR 9
115 Z8    WHILE lok%<lges%
116 6S4     IF lges%-lok%<puffl% THEN puffl%=lges%-lok%
117 xM      z$=INPUT$(puffl%,1)
118 LF      CALL CopyMem(SADD(z$),VARPTR(f%(0))+lok%,puffl%)
119 CI      lok%=lok%+puffl%
120 NB2   WEND
121 HA    z$=""
122 mW0 CLOSE 1
123 13  END SUB
124 I1  SUB FastSave (f%(),File$,puffl%) STATIC
125 IG  VL%=2
126 RD  IF puffl%<1 THEN puffl%=1024
127 Pb  OPEN File$ FOR OUTPUT AS 1
128 u52   lok%=0
129 2D    lges%=(UBOUND(f%)+1)*VL%
130 H1    z$=STRING$(puffl%,CHR$(0))
131 pO    WHILE lok%<lges%
132 M14     IF lges%-lok%<puffl% THEN puffl%=lges%-lok%
133 ih      CALL CopyMem(VARPTR(f%(0))+lok%,SADD(z$),puffl%)
134 iY      PRINT #1,LEFT$(z$,puffl%);
135 SY      lok%=lok%+puffl%
136 dR2   WEND
137 XQ    z$=""
138 2m0 CLOSE 1
139 HJ  END SUB
140 8Q  SUB Init (f%(),von%,bis%,wert%) STATIC
141 YW  VL%=2
142 J5  IF bis%=0 THEN bis%=UBOUND(f%)
143 mi  IF von%>bis% THEN SWAP von%,bis%
144 D6  Test=f%(von%)
145 CE  Test=f%(bis%)
146 Ue  l%=(bis%-von%+1)*VL%
147 cG  z$=MKI$(wert%)
148 OT  WHILE LEN(z$)*2<l%
149 zr2   z$=z$+z$
150 rf0 WEND
151 eI  z$=z$+LEFT$(z$,l%-LEN(z$))
152 JU  CALL CopyMem (SADD(z$),VARPTR(f%(von%)),l%)
153 ng  z$=""
154 WY  END SUB
155 T2  SUB Insert (f%(),ps%,wert%) STATIC
156 nl  VL%=2
157 ar  Test=f%(ps%)
158 Bu  IF ps%<UBOUND(f%) THEN
159 ae2   l%=(UBOUND(f%)-ps%+1)*VL%
160 Ib    z$=STRING$(l%,CHR$(0))
161 mu    CALL CopyMem(VARPTR(f%(ps%)),SADD(z$),l%)
162 3h    CALL CopyMem(SADD(z$),VARPTR(f%(ps%+1)),l%-VL%)
163 ZS0 END IF
164 yr  z$=""
165 M2  f%(ps%)=wert%
166 ik  END SUB
167 KQ  SUB Del (f%(),von%,bis%) STATIC
168 zx  VL%=2
169 kW  IF bis%=0 THEN bis%=UBOUND(f%)
170 D9  IF von%>bis% THEN SWAP von%,bis%
171 eX  Test=f%(von%)
172 df  Test=f%(bis%)
173 dx  ln%=(bis%-von%+1)*VL%
174 aH  l%=(UBOUND(f%)-von%+1)*VL%
175 Xq  z$=STRING$(l%,CHR$(0))
176 N9  CALL CopyMem(VARPTR(f%(von%)),SADD(z$),l%)
177 te  z$=MID$(z$,ln%+1)+STRING$(ln%,CHR$(0))
178 9G  CALL CopyMem(SADD(z$),VARPTR(f%(von%)),l%)
179 D6  z$=""
180 wy  END SUB
181 Ok  SUB Search (f%(),ab%,wert%,ps%) STATIC
182 DB  VL%=2
183 9H  Test=f%(ab%)
184 Sz  l%=(UBOUND(f%)+1)*VL%
185 BX  pp%=ab%*VL%
186 sH  z$=STRING$(l%,CHR$(32))
187 2p  CALL CopyMem(VARPTR(f%(0)),SADD(z$),l%)
188 Na  ok=0
189 5W  WHILE ok=0
190 132   pp%=INSTR(pp%+1,z$,MKI$(wert%))
191 BQ    ps=(pp%-1)/VL%
192 Av    IF ps=INT(ps) OR ps<0 THEN ok=1
193 YM0 WEND
194 oA  ps%=ps
195 P5  IF pp%=0 THEN ps%=-1
196 UN  z$=""
197 DF  END SUB
(C) 1989 M&T
```

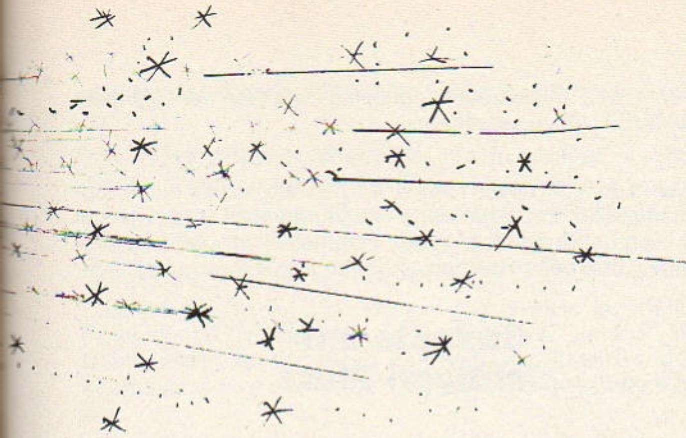
Listing. Machen Sie mit »Felder« Ihre Basic-Programme schneller. Bitte unbedingt mit dem Checksummer (Seite 143) eingeben.

Es ist soweit: 40 Zaubersprüche warten auf Sie. Eine Bibliothek voller Funktionen, leichtverständlich und einfach anzuwenden. Ob Basic, C oder Assembler, alle Programmierer haben Vorteile.

ZAUBERN

Vereinfachung von komplizierten und aufwendigen Programmen ist eines der Ziele der Bibliothek »extintui.library«. Dies trifft hauptsächlich für die C-Programmierer zu. Das andere Ziel ist, in Basic Dinge zu realisieren, die bisher nicht möglich waren. Beides ist mit der neuen Bibliothek (Library) gelungen. Der Anwendungsbereich erstreckt sich von Fenstern, über Pull-Down-Menüs mit Unterpunkten bis zur Auswertung von Intuition-Nachrichten. C-Programmierer öffnen wie gewohnt die Bibliothek mit der Funktion OpenLibrary(). Das Programm »extintui_lnk« müssen Sie zu Ihrem Hauptprogramm linken. Die Basic-Programmierer haben es einfacher. Sie konvertieren die Datei »extintui_lib.fd« (Listing 3) mit dem Programm »ConvertFD« (Basic-Diskette im Unterverzeichnis BasicDemos). Die erhaltene Datei »extintui.bmap« kopieren Sie zu Ihrem Basic-Programm. Im Listing 4 zeigen wir Ihnen den Vorspann, den Sie am Anfang Ihres Programms verwenden sollten. Damit das System bei eventuellen Aufräumarbeiten alles beseitigen kann, was nutzlos im Speicher steht, muß bei Programmende jede geöffnete Library mit CloseLibrary() geschlossen werden. Die Funktionen dienen zur vereinfachten Steuerung der Benutzeroberfläche. Speziell als Anfänger auf dem Amiga ist es schwer, mit den komplexen Strukturen umzugehen, die nötig sind, um ein Fenster zu öffnen oder eine Menüleiste erscheinen zu lassen. Die Arbeit wird nun komplett von der Library übernommen. Die Funktion ShowWindow() öffnet ein Fenster mit der angegebenen Größe entweder auf dem Workbench-Screen oder einem zuvor geöffneten eigenen Bildschirm. Öffnet man nacheinander ein Fenster, einen Bildschirm und wieder ein Fenster, steht das erste Fenster auf dem Workbench-Screen und das zweite in dem zuvor geöffneten Bildschirm. Wird mit den Zeichenfunktionen der Library ein Kreis gezeichnet, so erscheint dieser immer in dem zuletzt geöffneten Fenster. Ebenso verhält es sich mit den Menü- und Gadget-Funktionen. Zusätzlich bietet die Bibliothek noch einige Fähigkeiten, die bisher von Amiga-Basic aus nicht

Zeichnung: Jan Valach



benutzt werden konnten. So etwa Menüunterpunkte und Requester. Doch zusammen mit anderen Funktionen sind die Wirkungen etwas unterschiedlich. So kann zum Beispiel keine Zeichenfunktion ausgeführt werden, wenn zuvor ein Bildschirm oder Fenster geschlossen wurde. Beim Löschen, werden die Adressen des aktuellen Bildschirms und Fensters, die in einem eigens angefertigten Puffer gespeichert werden, ebenfalls gelöscht. Versucht nun eine Zeichenfunktion, auf dem aktuellen Fenster etwas auszugeben, scheitert die Ausführung, da keine Fenster-Adresse gefunden werden kann. Und nun wollen wir uns den einzelnen Funktionen widmen.

```
screen = ShowScreen(Tiefe,Modus,Titel)
```

SIE MIT

Mit dieser Funktion ist es sehr einfach, einen Screen auf dem Monitor darzustellen. Der Parameter »Tiefe« gibt die Anzahl der Bit-Ebenen und dadurch automatisch die möglichen Farben an. Im Zuge der Vereinfachung wurden für den Parameter »Modus« vier Möglichkeiten festgelegt, die Sie beliebig erweitern oder verändern können:

- 0 = Normalauflösung mit 320 x 256 Pixeln in maximal 64 Farben, was einer Tiefe von 6 entspricht.
- 1 = Hi-Res-Screen mit 640 x 256 Pixeln in maximal 16 Farben und einer Tiefe von maximal 4.
- 2 = Lace-Screen mit 320 x 512 Pixeln in maximal 32 Farben und einer Maximaltiefe von 5.
- 3 = Hi-Res und Lace mit 640 x 512 Pixeln in maximal 16 Farben bei einer Tiefe von 4.

Der Parameter »Titel« enthält den Namen des Bildschirms, der in der Titelleiste erscheint. Die Funktion liefert bei Erfolg die Adresse des Bildschirms zurück. Sie wird benötigt, um den Bildschirm später wieder schließen zu können. Sobald der Screen erscheint, ist er der für das System aktuelle Bildschirm; alle Fensterfunktionen aus der Library beziehen sich auf diesen Screen.

```
SetPal (Farbregister, rot, grün, blau)
```

Mit dieser Funktion setzen Sie die einzelnen Farbregister für Ihren Bildschirm. »Farbregister« ist die Nummer der gewünschten Farbe. Die anderen Parameter können Werte von 0 bis 15 annehmen. 15 bedeutet dabei volle Stärke.

```
ActScreen(screen)
```

Wurden mehrere Screens geöffnet, kann mit dieser Funktion einer ausgewählt werden, der zum aktuellen wird. Der Parameter »screen« ist die Adresse des Bildschirms, der von der Funktion ShowScreen() zurückgeliefert wird.

```
RemScreen(screen)
```

ERWEITERUNG

Die Funktion RemScreen() ruft Close Screen() auf, durchläuft zuvor allerdings einige Routinen, die alle auf dem Bildschirm vorhandenen Fenster löscht. Durch diese Funktion werden die Adressen für das aktuelle Fenster und den Bildschirm zurückgesetzt.

```
window =ShowWindow(LeftEdge, TopEdge,
Width, Height, IDCMP, Flags, Titel)
```

Zum Erzeugen von Fenstern dient die Funktion ShowWindow(). Sie ist ebenfalls so einfach anzusprechen wie die Funktion ShowScreen(). »LeftEdge« und »TopEdge« geben die linke obere Ecke des Fensters auf dem Bildschirm an. »Width« und »Height« sind die Breite und die Höhe. Die Koordinaten werden überprüft und gegebenenfalls korrigiert. Wird beispielsweise ein Bildschirm in der Größe 320 x 200 Pixel geöffnet, ist es unmöglich, danach ein Fenster in der Größe 600 x 250 Pixel zu öffnen. Nach den Koordinaten werden die IDCMP-Flags für das entsprechende Fenster erwartet. Die IDCMP-Flags, wie auch die Window-Flags, sollen nicht alle aufgelistet werden. Informationen darüber finden Sie in unserem C-Kurs (AMIGA-Magazin, Ausgabe 7/88, Seite 97) und praktisch allen Programmierhandbüchern über den Amiga. »Titel« ist die Adresse eines mit einem Null-Byte abgeschlossenen Strings. Basic-Programmierer müssen daher darauf achten, daß sie ihre Basic-Strings mit einem Null-Byte versehen und die Adresse des Strings an die Funktionen übergeben und nicht etwa den String-deskriptor. Zuerst muß die 0 angehängt werden, was folgendermaßen geschieht a$=a$+CHR$(0). Dann verwenden Sie den Befehl SADD, der die Adresse der Zeichenkette berechnet.

```
Act(Window(window)
```

ActWindow() erklärt ein Fenster als aktuell. Aktualisieren bedeutet, daß alle folgenden Zeichenfunktionen auf das in »window« angegebene Fenster umgelenkt werden. Das Fenster wird nicht in den Vordergrund gestellt.

```
RemWindow(window)
```

Diese Funktion dient zum Löschen einzelner Fenster. Sollte das Fenster das aktuelle sein, wird die Adresse innerhalb des speziellen Datenbereichs auf Null gesetzt. Zeichenfunktionen können dann erst wieder nach einem ShowWindow()- oder ActWindow()-Aufruf ausgeführt werden. Die Zeichen-, Textausgabe- und Nachfragefunktionen der Library beziehen sich, wie bereits erwähnt, auf das jeweils aktuelle Fenster. Deshalb wird bei keiner dieser Funktionen die Fenster- oder RastPort-Adresse verlangt. Sie müssen lediglich darauf achten, daß ein aktuelles Fenster vorhanden ist. Sollen Ausgaben in verschiedenen, bereits geöffneten Fenstern stattfinden, müssen die jeweiligen Fenster vor der Ausgabe mit ActWindow() aktualisiert werden.

```
PutCircle(x,y,radius)
PutEllipse(x,y,vradius,hradius)
```

PutCircle() zeichnet einen Kreis und PutEllipse() eine Ellipse in das aktuelle Fenster. »x« und »y« sind dabei jeweils der Mittelpunkt. »vradius« ist der vertikale und »hradius« der horizontale Radius der Ellipse. Bei einem Kreis sind beide Werte gleich, daher muß nur der Radius im Parameter »radius« angegeben werden.

```
PutPoint(x,y)
```

Setzt an der Position mit den Koordinaten »x« und »y« einen Punkt in der aktuellen Farbe.

```
PutLine(x1,y1,x2,y2)
```

PutLine() zieht eine Linie vom Punkt »x1« und »y1« zu den Koordinaten »x2« und »y2«. Es wird die aktuelle Farbe verwendet.

```
PrintText(Text,x,y)
```

Zur Grafiktextausgabe wird die Funktion PrintText() eingesetzt. Der Text wird dabei in der aktuellen Zeichenfarbe und dem Modus ausgegeben. Die Position wird wieder mit »x« und »y« festgelegt.

```
SetColor(front,back,out)
Mode(Drawmode)
```

Die Funktionen SetColor() und Mode() schreiben die angegebenen Werte für Farben und Zeichenmodi direkt in den RastPort des aktuellen Fensters. Wechseln Sie mit ActWindow() das aktuelle Fenster, müssen die Farb- und Zeichenmodi neu eingestellt werden, wenn dies nicht bereits geschah. »front« und »back« sind die Vorder- und Hintergrundfarbe. Mit »out« geben Sie die Farbe für Umrahmungen an.

```
x=GetPosX()            y=GetPosY()
```

Die GetPos()-Funktionen ermitteln die aktuelle Position des Grafik-Cursors innerhalb des aktuellen Fensters. Basic-Programmierer müssen hier wie bei den anderen Funktionen, die keinen Parameter erwarten, eine »0« einsetzen, da sonst Amiga-Basic einen Syntax-Fehler anzeigt. Der Grafik-Cursor wird beim Zeichnen verwendet. Es werden zwei getrennte Funktionen verwendet, da sonst von Basic aus wieder mit Zeigern gearbeitet werden müßte. Dasselbe gilt für die folgenden zwei Funktionen:

```
x=MouseX()            x=MouseY()
```

MouseX() und MouseY() geben Auskunft über die Position des Mauszeigers innerhalb des Fensters. Das Erzeugen von Amiga-Menüleisten ist in C eine komplizierte Sache. Es müssen Strukturen versorgt und miteinander verbunden werden. Die Menüfunktionen der Library machen dies automatisch und müssen nur in der Reihenfolge, in der das Menü angezeigt werden soll, aufgerufen werden.

Die Größe des Selektierungsbereichs der einzelnen Menüeinträge richtet sich nach dem längsten im gesamten Menü enthaltenen Text. Somit ist gewährleistet, daß alle Menüeinträge in der gleichen Größe erscheinen. Alle Menüfunktionen mit Ausnahme von KillMenu() beziehen sich auf das aktuelle Fenster. Ist kein Fenster vorhanden, auf dem eine Menüleiste angezeigt werden kann, werden die Funktionen erfolglos abgebrochen.

```
MenuHeader(text)
```

erzeugt eine neue Menüleiste. Ist bereits eine Menüleiste für das aktuelle Fenster generiert worden, wird die neue Menüleiste rechts daneben angezeigt.

```
MenuItem(text)
```

stellt einen neuen Menüeintrag unter eine Menüleiste oder einen bereits vorhandenen Menüeintrag. Ein Menüeintrag kann nur erzeugt werden, wenn eine Menüleiste vorhanden ist.

```
MenuSubItem(text)
```

Untereinträge werden immer an zuletzt erzeugte Untereinträge oder Menüeinträge angehängt. Diese Funktion kann nicht ausgeführt werden, wenn die Menüleiste keinen Menüeintrag enthält oder keine Menüleiste vorhanden ist.

```
KillMenu(window)
```

Hat eine Menüleiste ihren Dienst getan, muß die Funktion KillMenu() aufgerufen werden. Der von der Menüleiste belegte Speicherplatz wird dann dem System wieder zur Verfügung gestellt. Die Menüleiste ist nicht mehr anwählbar. Unabhängig vom aktuellen Fenster muß die Fenster-Adresse mitgeteilt werden.

Fenster, Gadgets und Requester

Komplizierter als Menüleisten sind Gadgets und Requester. Wo bei Menüleisten nur zwei Strukturen versorgt werden müssen, sind es bei Gadgets gleich vier. Dies liegt an der vielseitigen Gestaltungs- und Anwendungsmöglichkeit von Gadgets. Bei Boolean-Gadgets ist dies vielleicht noch recht einfach, bei Proportional- oder String-Gadgets müssen zusätzlich noch Info-Strukturen angelegt werden. In der Library besorgen dies die Funktionen SetPropInfo() und SetStringInfo(), die das Ergebnis »Info« liefern. Der Parameter »Info« wird dann der SetGadget()-Funktion übergeben. Genau wie bei den Menüleisten, müssen Sie je nach gewünschter Anzahl der Gadgets die Funktion SetGadget() aufrufen. Erst nach Öffnen eines Requesters mit der Funktion ShowRequester() werden die zuvor mit SetGadget() erzeugten Gadgets im aktuellen Fenster angezeigt. Die Reihenfolge, in der die Funktionen aufgerufen werden müssen, lautet wie folgt:

1. Fenster öffnen mit ShowWindow()
2. Gadgets erzeugen mit SetGadget()
3. Requester öffnen mit ShowRequester()

```
strinfo = SetStringInfo(maxchars)
```

Reserviert Speicherplatz für eine StringInfo-Struktur und liefert deren Adresse als Ergebnis. Der Parameter »maxchars« legt die Anzahl der Zeichen fest, die in das Stringfeld eingegeben werden können. Die zurückerhaltene Adresse muß der Funktion SetGadget() im Parameter »Info« mitgeteilt werden. Außerdem ist der Parameter »Typ« der Funktion SetGadget() mit dem Wert der Definition »STRGADGET« (4) zu versorgen.

```
propinfo = SetPropInfo(Flags,hpot,vpot,hbody,vbody)
```

Die Funktion SetPropInfo reserviert Speicherplatz für eine Prop-Info-Struktur. Die zurückgelieferte Strukturadresse muß, wie bei der Funktion SetStringInfo(), an den Parameter »Info« der Funktion SetGadget() übergeben werden. Im Parameter »Flags« wird mitgeteilt, ob sich der Slider in vertikaler (4), horizontaler (2) oder in beiden Richtungen (6) bewegen läßt. Die Parameter »hpot« und »vpot« bestimmen die Position des Reglers innerhalb des Gadgets. Genau wie »hbody« und »vbody«, können die Parameter »hpot« und »vpot« Werte von 0 (klein) bis 65335 (groß) annehmen.

Gadgets am richtigen Platz

```
gadget =SetGadget(LeftEdge,TopEdge,Width,
Height,Activ,Typ,Text,Info,Id)
```

Mit dieser Funktion werden die Gadgets erzeugt, aber noch nicht auf dem Bildschirm angezeigt. Die Größe und Position des Gadgets innerhalb des Requesters, wird mit den Parametern »LeftEdge«, »TopEdge«, »Width« und »Height« festgelegt. Sie haben dieselbe Bedeutung wie bei ShowWindow(). In »Activ« wird angegeben, welche Nachricht das Gadget liefern soll, wenn es angeklickt wird, oder speziell bei String-Gadgets, wie der eingegebene Text im Stringfeld erscheinen soll.

RELVERIFY (1) > GADGETUP
GADGIMMEDIATE (2) > GADGETDOWN

Diese beiden »Activ«-Werte müssen gesetzt sein, wenn das Gadget eine IntuiMessage erzeugen soll, sobald es mit dem Mauszeiger angeklickt wird. Das Fenster, in dem der Requester mit den Gadgets angezeigt wird, muß dazu in den IDCMP-Flags die Werte für GADGETDOWN oder GADGETUP enthalten.

Für String-Gadgets können zusätzlich noch folgende Werte gesetzt werden:

- STRCENTER (512) : Die Eingabe erfolg zentriert.
- STRRIGHT (1024) : Die Eingabe erfolgt rechtsbündig.
- LONGINT (2048) : Es können nur Ziffern eingegeben werden.

Der Parameter »Typ« muß Informationen über die Art des Gadgets enthalten. Bei einem Boolean-Gadget den Wert 1 für »BOOL GADGET«. Der Parameter »Info« wird für Boolean-Gadgets nicht berücksichtigt. Für String-Gadgets muß der Parameter »Typ« den Wert 4 für »STRGADGET«, und der Parameter »Info« die Adresse der reservierten StringInfo-Struktur, enthalten. Bei Proportional- Gadgets wird in »Typ« der Wert 3 für »PROPGADGET« und in »Info« die Adresse einer PropInfo-Struktur erwartet.

»Text« ist ein mit einem Null-Byte beendeter String, der im Gadget angezeigt wird. In Boolean-Gadgets wird der Text innerhalb des Gadgets zentriert ausgegeben. Bei Proportional-Gadgets steht der Text über dem Gadget, da der Slider ihn überdecken würde. Bei String-Gadgets steht er links neben dem Eingabefeld.

Die Kennziffer oder Id des Gadgets wird im Parameter »id« erwartet. Die Gadget-Kennung ist wichtig, um bei einer aufgetretenen Nachricht zu erfahren, welches Gadget angeklickt wurde. Tritt eine GADGETDOWN- oder GADGETUP-Nachricht ein, kann mit der Funktion Gadgetid() die Kennung des angeklickten Gadgets erfragt und entsprechend reagiert werden.

```
textadr = GetStrinfo(requester,id)
```

Erhält man die GADGETUP-Nachricht von einem String-Gadget, liefert die Funktion GetStrinfo() die Adresse des eingegebenen Strings zurück. Der Parameter »requester« wird von der Funktion Showrequester() geliefert und muß zur Identifizierung des Gadgets angegeben werden. »id« muß die Kennung des Gadgets enthalten.

Die nachfolgenden Funktionen benötigen, gleich der Funktion GetStrinfo(), die Parameter »requester« und »id« zur genauen Lokalisierung des Gadgets.

```
x = GetPropX(requester,id)
y = GetPropY(requester,id)
```

GetPropX() und GetPropY() liefern die Position des Reglers innerhalb eines Proportional-Gadgets. Achten Sie bitte darauf, daß Sie die Variablen für den Empfang der Werte so definieren, daß der Wertebereich von 0 bis 65335 reicht. Basic-Programmierer ver-

knüpfen das Ergebnis in diesem Fall logisch mit 65535 (x% AND 65535&).

```
w = GetPropW(requester,id)
h = GetPropH(requester,id)
```

Die beiden Funktionen liefern die Größe des Reglers im Bereich 0 bis 65535.

```
ChangeProp(requester,id,hpot,vpot,hbody,vbody)
```

Damit die Größe und Position des Reglers innerhalb eines Proportional-Gadgets geändert werden kann, steht die Funktion ChangeProp() zur Verfügung. Zur Requester-Adresse und Gadget-Kennung müssen die neuen Ausmaße des Sliders angegeben werden.

```
requester = ShowRequester(LeftEdge, TopEdge,
Width, Height, Farbe)
```

ShowRequester() öffnet einen Requester in der angegebenen Größe und Farbe im aktuellen Fenster. Wurden zuvor Gadgets erzeugt, werden diese im Requester mit angezeigt. Hatte die Funktion Erfolg, wird die Adresse des Requesters zurückgeliefert. Sie wird benötigt, um den Requester wieder zu schließen. Kann kein Requester geöffnet werden, wird eine Null zurückgeliefert.

```
RequesterEnd(requester)
```

Die Funktion RequesterEnd() läßt den Requester wieder verschwinden und gibt den durch die Gadgets belegten Speicherplatz frei. Soll der gleiche Requester erneut geöffnet werden, müssen alle darin enthaltenen Gadgets neu erzeugt werden.

Der Nachrichtenempfang bezieht sich immer auf ein Fenster. Beim Öffnen eines Fensters muß deshalb in den IDCMP-Flags mitgeteilt werden, welche Nachrichten das jeweilige Fenster empfangen kann. Wird beispielsweise der Wert für MOUSEBUTTONS an das IDCMP-Flag des Fensters übergeben, erhält man immer eine Nachricht, wenn eine der Maustasten gedrückt wird.

```
ok = Message(window)
```

Message() liefert 1, wenn für das angegebene Fenster eine Nachricht aufgetreten ist, andernfalls eine 0. Ist eine Nachricht eingegangen, kann mit der Funktion Class() die Art (IDCMP) der Nachricht erfragt werden. Die Funktion sollte in einer Schleife aufgerufen werden, die erst dann wieder verlassen wird, wenn ein bestimmtes Ereignis eintritt.

```
class = Class()
```

Class() liefert den Class-Wert der letzten Nachricht.

```
code = Code()
```

Code() gibt den Code-Wert der letzten Nachricht zurück.

```
id = Gadgetid()
```

Enthält der Class-Wert der eingetretenen Nachricht den Wert GADGETDOWN oder GADGETUP, ermittelt die Funktion Gadgetid() die Kennung des angeklickten Gadgets.

Ist die Nachricht MENUPICK eingetreten, kann mit den nachfolgenden Funktionen der angewählte Menüeintrag oder Untereintrag ermittelt werden:

```
id = Menuid()
```

Liefert die Kennung der angewählten Menüleiste.

```
id = Itemid()
```

Liefert die Kennung des angewählten Menüeintrags.

```
id = Subid()
```

Liefert die Kennung des angewählten Untermenüeintrags.

Die Routinen für Funktionen

Das Prinzip, in Form von Bibliotheken (Libraries) Routinen zur Programmierung bereitzustellen, gehört zum Amiga wie das Salz in die Suppe. Um den großen Aufwand der Vorbereitung und Verwaltung zu verkleinern, gibt es jetzt unsere Bibliothek »extintui.library«. Kurze Programme mit großer Wirkung sind kein Wunschtraum mehr.

Bisher fanden Sie die genaue Beschreibung der einzelnen Funktionen. Was noch fehlt, sind die Routinen selbst. Die finden Sie in Listing 5. Zusammen mit den Listings aus der letzten Ausgabe können Sie nun die Bibliothek generieren. Falls Sie den benötigten Aztec-C-Compiler nicht besitzen, finden Sie alle Programme in lauffähiger Version auf der Programmservice-Diskette.

Für Basic-, C- und Assembler-Programmierer war bis jetzt der Zugriff auf die 40 Funktionen der neuen Bibliothek »extintui.library« möglich. Damit auch die Modula-2-Fans in den Genuß der einfachen Programmierung kommen, finden Sie im Listing 6 das nötige Modula-2-Programm. Compilieren Sie es und kopieren Sie die generierte Datei »M2Extintui.sym« in das Verzeichnis »Modules/Interfaces/sym« Ihrer Modula-2-Systemdiskette.

Die formalen Parameter der Prozeduren sind gemäß der in »Intuition.def« und »Graphics.def« festgelegten Typendeklarationen benannt. Dadurch wird der eventuelle Umstieg in die direkte Programmierung von Intuition erleichtert. Das Umsetzen des Demoprogramms dürfte ein Kinderspiel sein.

Und nun Start frei für kurze, benutzerfreundliche Programme in vier Programmiersprachen. *Jürgen Haage/Andreas Görtler/kn*

Compiler- und Assembleranweisungen für Listing 1:
as -o extintui__lib.o extintui__lib.asm -D -N
as -o extintui__lnk.o extintui__lnk.asm -D -N

Programmname:	extintui__lib.asm
Computer:	A500, A1000, A2000 mit Kickstart 1.2, 1.3
Sprache:	Assembler
Bemerkung:	siehe Text

```
Programmautor: Jürgen Haage
---------------------------

7b0 */************************************/
wq  */* extintui_lib.c  v1.0       */
1J  */* Jürgen Haage  Burgstraße 30       */
VT  */* 6552 Bad Münster a. St. Ebernburg 2 */
Bf  */************************************/
CD  VERSION    equ   1
KC  REVISION   equ   1
0w3    nolist
rp     include "exec/types.i"
Hn     include "exec/initializers.i"
kF     include "exec/libraries.i"
JF     include "exec/lists.i"
1U     include "exec/resident.i"
vm     include "exec/strings.i"
Vn     include "libraries/dos.i"
JD     include "exec/io.i"
OU     list
Ke     STRUCTURE   ExtintuiBase,LIB_SIZE
FM     RPTR   extin_SegList
a0     APTR   extin_SysBase
Eo     APTR   extin_IntuitionBase
yV8        APTR   extin_GfxBase
BB3    LABEL   extin_SIZEOF
CX0 EXTINTUILIBNAME   macro
BX3    dc.b   'extintui.library',0
1j     endm
9b0 xlib   macro
303    xref   _LVO%1
4m     endm
xY0 callsys   macro
9p3    jsr   _LVO%1(a6)
7p     endm
YR     xref   _FindLib
yd     xref   _ShowWindow
bY     xref   _ShowScreen
UC     xref   _ActScreen
qa     xref   _RemScreen
Rc     xref   _ActWindow
n0     xref   _RemWindow
iR     xref   _PutCircle
N6     xref   _PutEllipse
Gb     xref   _PutLine
2E     xref   _PutPoint
OP     xref   _SetColor
VL     xref   _Mode
m5     xref   _GetPosX
t8     xref   _GetPosY
aJ     xref   _PrintText
17     xref   _MouseX
rE     xref   _MouseY
zz     xref   _MenuHeader
K2     xref   _MenuItem
LY     xref   _MenuSubItem
Gs     xref   _KillMenu
8j     xref   _SetStringInfo
5L     xref   _SetPropInfo
7V     xref   _SetGadget
0z     xref   _GetStrinfo
```

Listing 1. »extintui__lib.asm« ist der Rumpf der Bibliothek

```
        xref    _GetPropX
        xref    _GetPropY
        xref    _GetPropW
        xref    _GetPropH
        xref    _ChangeProp
        xref    _ShowRequester
        xref    _RequesterEnd
        xref    _Message
        xref    _Class
        xref    _Code
        xref    _Gadgetid
        xref    _Menuid
        xref    _Itemid
        xref    _Subid
        xref    _SetPal
        xlib    OpenLibrary
        xlib    CloseLibrary
        xlib    FindTask
        xlib    Remove
        xlib    FreeMem
        xdef    _SysBase
        xdef    _IntuitionBase
        xdef    _GfxBase
        public  _AllocMem
        CSEG
        moveq   #RETURN_FAIL,d0
        rts
ROMTag:
        dc.w    RTC_MATCHWORD
        dc.l    ROMTag
        dc.l    EndCode
        dc.b    RTF_AUTOINIT
        dc.b    VERSION
        dc.b    NT_LIBRARY
        dc.b    0
        dc.l    libraryName
        dc.l    idString
        dc.l    Init
libraryName:
        EXTINTUILIBNAME
idString:
        dc.b    'extintuilib 1.0 (12.OKT 1988)',CR,LF,0
intuilib:
        dc.b    'intuition.library',0
        ;cnop   0,2
gfxlib:
        dc.b    'graphics.library',0
        ;cnop   0,2
Init:
        dc.l    extin_SIZEOF
        dc.l    funcTable
        dc.l    dataTable
        dc.l    .InitLib
funcTable:
        dc.l    .OpenLib
        dc.l    .CloseLib
        dc.l    .ExpungeLib
        dc.l    .ExtFuncLib
        dc.l    .FindLib
        dc.l    .ShowWindow
        dc.l    .ShowScreen
        dc.l    .ActScreen
        dc.l    .RemScreen
        dc.l    .ActWindow
        dc.l    .RemWindow
        dc.l    .PutCircle
        dc.l    .PutEllipse
        dc.l    .PutLine
        dc.l    .PutPoint
        dc.l    .SetColor
        dc.l    .Mode
        dc.l    .GetPosX
        dc.l    .GetPosY
        dc.l    .PrintText
        dc.l    .MouseX
        dc.l    .MouseY
        dc.l    .MenuHeader
        dc.l    .MenuItem
        dc.l    .MenuSubItem
        dc.l    .KillMenu
        dc.l    .SetStringInfo
        dc.l    .SetPropInfo
        dc.l    .SetGadget
        dc.l    .GetStrinfo
        dc.l    .GetPropX
        dc.l    .GetPropY
        dc.l    .GetPropW
        dc.l    .GetPropH
        dc.l    .ChangeProp
        dc.l    .ShowRequester
        dc.l    .RequesterEnd
        dc.l    .Message
        dc.l    .Class
        dc.l    .Code
        dc.l    .Gadgetid
        dc.l    .Menuid
        dc.l    .Itemid
        dc.l    .Subid
        dc.l    .SetPal
        dc.l    -1
dataTable:
        INITBYTE    LH_TYPE,NT_LIBRARY
        INITLONG    LN_NAME,libraryName
        INITBYTE    LIB_FLAGS,LIBF_SUMUSED!LIBF_CHANGED
        INITWORD    LIB_VERSION,VERSION
        INITWORD    LIB_REVISION,REVISION
        INITLONG    LIB_IDSTRING,idString
        dc.l    0
.InitLib:
        move.l  a5,-(sp)
        movea.l d0,a5
        move.l  a0,extin_SegList(a5)
        lea     intuilib(pc),a1
        moveq   #LIBRARY_VERSION,d0
        callsys OpenLibrary
        tst.l   d0
        beq.s   1$
        move.l  a6,extin_SysBase(a5)
        move.l  a6,_SysBase
        move.l  d0,extin_IntuitionBase(a5)
        move.l  d0,_IntuitionBase
        lea     gfxlib(pc),a1
        moveq   #LIBRARY_VERSION,d0
        callsys OpenLibrary
        tst.l   d0
        beq     1$
            move.l  d0,extin_GfxBase(a5)
        move.l  d0,_GfxBase
        move.l  a5,d0
1$
        movea.l (sp)+,a5
        rts
.OpenLib:
        addq.w  #1,LIB_OPENCNT(a6)
        bclr    #LIBB_DELEXP,LIB_FLAGS(a6)
        move.l  a6,d0
        movem.l d0/a1/a5,-(sp)
        move.l  a6,a5
        move.l  16(sp),a6
        pea     65536       MEMF_CLEAR
        pea     100
        jsr     _AllocMem    Speicher anfordern
        add.w   #8,sp
        tst.l   d0
        beq     1$
        move.l  d0,-(sp)
        move.l  #0,a1
        callsys FindTask    Taskadresse ermitteln
        move.l  d0,a1
        move.l  (sp)+,d0
        move.l  d0,88(a1)   Adresse des Speicherbereiches
        move.l  a5,a6
        movem.l (sp)+,d0/a1/a5
        rts
1$
        movem.l (sp)+,d0/a1/a5
        move.l  #0,d0
        rts
.CloseLib:
        movem.l a1-a2/a5,-(sp)
        move.l  a6,a5
        move.l  extin_SysBase(a5),a6
00$     sub.l   a1,a1
        callsys FindTask
        tst.l   d0
        beq.s   00$
        move.l  d0,a2 ;Taskadresse ermitteln
        move.l  88(a2),d0
        beq.s   01$
        move.l  d0,a1
        move.l  #100,d0
        callsys FreeMem   ;user Speicher wieder freigeben
01$     movem.l (sp)+,a1-a2/a5
        move.l  a5,a6
        moveq   #0,d0
        subq.w  #1,LIB_OPENCNT(a6)
        bne.s   1$
        btst    #LIBB_DELEXP,LIB_FLAGS(a6)
        beq.s   1$
        bsr.s   .ExpungeLib
1$      rts
.ExpungeLib:
        movem.l d2/a5/a6,-(sp)
        tst.w   LIB_OPENCNT(a6)
        beq.s   1$
        bset    #LIBB_DELEXP,LIB_FLAGS(a6)
        moveq   #0,d0
        bra.s   2$
1$
        move.l  a6,a5
        move.l  extin_SysBase(a5),a6
        movea.l extin_IntuitionBase(a5),a1
        callsys CloseLibrary
        movea.l extin_GfxBase(a5),a1
        callsys CloseLibrary
        move.l  a5,a1
        callsys Remove
        move.l  extin_SegList(a5),d2
        movea.l a5,a1
        moveq   #0,d0
        move.w  LIB_NEGSIZE(a5),d0
        suba.l  d0,a1
        add.w   LIB_POSSIZE(a5),d0
        callsys FreeMem
        move.l  d2,d0
2$
        movem.l (sp)+,d2/a5/a6
        rts
.ExtFuncLib:
        moveq   #0,d0
        rts
.FindLib
        jsr     _FindLib(pc)
        rts
.ShowWindow
        move.l  a0,-(a7)
        movem.l d0-d5,-(a7)
        jsr     _ShowWindow(pc)
        adda.w  #28,sp
        rts
.ShowScreen
        move.l  a0,-(a7)
        movem.l d0-d1,-(a7)
        jsr     _ShowScreen(pc)
        adda.w  #12,sp
        rts
.ActScreen
        move.l  a0,-(a7)
        jsr     _ActScreen(pc)
        addq.l  #4,sp
        rts
.RemScreen
        move.l  a0,-(a7)
        jsr     _RemScreen(pc)
        addq.l  #4,sp
        rts
.ActWindow
```

```
Lb3    move.l   a0,-(a7)
IO     jsr   _ActWindow(pc)
F4     addq.l   #4,sp
tu     rts
020  .RemWindow
Qg3    move.l   a0,-(a7)
xQ     jsr   _RemWindow(pc)
K9     addq.l   #4,sp
yz     rts
nc0  .PutCircle
GV3    movem.l  d0-d2,-(a7)
XP     jsr   _PutCircle(pc)
4b     adda.w   #12,sp
34     rts
AB0  .PutEllipse
Rc3    movem.l  d0-d3,-(a7)
1N     jsr   _PutEllipse(pc)
Lw     adda.w   #16,sp
89     rts
x00  .PutLine
Wh3    movem.l  d0-d3,-(a7)
fW     jsr   _PutLine(pc)
Q1     adda.w   #16,sp
DE     rts
JQ0  .PutPoint
P13    movem.l  d0-d1,-(a7)
h0     jsr   _PutPoint(pc)
mf     addq.l   #8,sp
IJ     rts
i40  .SetColor
ap3    movem.l  d0-d2,-(a7)
LQ     jsr   _SetColor(pc)
Ov     adda.w   #12,sp
NO     rts
D50  .Mode
CG3    move.l   d0,-(a7)
OW     jsr   _Mode(pc)
od     addq.l   #4,sp
ST     rts
nX0  .GetPosX
nH3    jsr   _GetPosX(pc)
VW     rts
uf0  .GetPosY
vQ3    jsr   _GetPosY(pc)
YZ     rts
ug0  .PrintText
k33    movem.l  d0-d1,-(a7)
6M     move.l   a0,-(a7)
c2     jsr   _PrintText(pc)
fC     adda.w   #12,sp
ef     rts
9P0  .MouseX
TP3    jsr   _MouseX(pc)
hi     rts
FW0  .MouseY
aX3    jsr   _MouseY(pc)
kl     rts
vX0  .MenuHeader
HX3    move.l   a0,-(a7)
NW     jsr   _MenuHeader(pc)
B0     addq.l   #4,sp
pq     rts
Qd0  .MenuItem
Mc3    move.l   a0,-(a7)
ul     jsr   _MenuItem(pc)
G5     addq.l   #4,sp
uv     rts
rY0  .MenuSubItem
Rh3    move.l   a0,-(a7)
dC     jsr   _MenuSubItem(pc)
LA     addq.l   #4,sp
z0     rts
LE0  .KillMenu
Wm3    move.l   a0,-(a7)
Vp     jsr   _KillMenu(pc)
QF     addq.l   #4,sp
45     rts
d10  .SetStringInfo
br3    move.l   a0,-(a7)
z1     jsr   _SetStringInfo(pc)
VK     addq.l   #4,sp
9A     rts
Sn0  .SetPropInfo
dk3    movem.l  d0-d4,-(a7)
s8     jsr   _SetPropInfo(pc)
Bh     adda.w   #20,sp
EF     rts
1k0  .SetGadget
BK3    movem.l  a0-a2,-(a7)
ps     movem.l  d0-d5,-(a7)
go     jsr   _SetGadget(pc)
bE     adda.w   #36,sp
KL     rts
mg0  .GetStrinfo
9D3    move.l   d0,-(a7)
s8     move.l   a0,-(a7)
X1     jsr   _GetStrinfo(pc)
un     addq.l   #8,sp
QR     rts
Jk0  .GetPropX
FJ3    move.l   d0,-(a7)
yE     move.l   a0,-(a7)
2v     jsr   _GetPropX(pc)
0t     addq.l   #8,sp
WX     rts
uw0  .GetPropY
LP3    move.l   d0,-(a7)
4K     move.l   a0,-(a7)
E3     jsr   _GetPropY(pc)
6z     addq.l   #8,sp
cd     rts
qq0  .GetPropW
RV3    move.l   d0,-(a7)
AQ     move.l   a0,-(a7)
85     jsr   _GetPropW(pc)
C5     addq.l   #8,sp
1j     rts
JU0  .GetPropH
Xb3    move.l   d0,-(a7)
GW     move.l   a0,-(a7)
mh     jsr   _GetPropH(pc)
IB     addq.l   #8,sp
op     rts
uQ0  .ChangeProp
IP3    movem.l  d0-d4,-(a7)
Mc     move.l   a0,-(a7)
HU     jsr   _ChangeProp(pc)
3d     adda.w   #24,sp
uv     rts
js0  .ShowRequester
OV3    movem.l  d0-d4,-(a7)
9K     jsr   _ShowRequester(pc)
wS     adda.w   #20,sp
z0     rts
X50  .RequesterEnd
Wm3    move.l   a0,-(a7)
Mv     jsr   _RequesterEnd(pc)
QF     addq.l   #4,sp
45     rts
oA0  .Message
br3    move.l   a0,-(a7)
RR     jsr   _Message(pc)
VK     addq.l   #4,sp
9A     rts
690  .Class
yM3    jsr   _Class(pc)
CD     rts
YG0  .Code
AW3    jsr   _Code(pc)
FG     rts
ZJ0  .Gadgetid
zC3    jsr   _Gadgetid(pc)
IJ     rts
Ub0  .Menuid
fB3    jsr   _Menuid(pc)
LM     rts
oT0  .Itemid
Xg3    jsr   _Itemid(pc)
OP     rts
KC0  .Subid
1L3    jsr   _Subid(pc)
RS     rts
xe0  .SetPal
p03    movem.l  d0-d3,-(a7)
JM     jsr   _SetPal(pc)
JK     adda.w   #16,sp
WX     rts
VY0  EndCode:
6B3    GLOBAL   _SysBase,4
CZ     GLOBAL   _IntuitionBase,4
9o     GLOBAL   _GfxBase,4
v0     end
(C) 1988 M&T
```

Listing 1. (Schluß)

Programmname:	extintui_lnk.asm
Computer:	A500, A1000, A2000 mit Kickstart 1.2, 1.3
Sprache:	Assembler
Bemerkung:	siehe Text

```
Programmautor: Jürgen Haage
---------------------------
7b0  */*******************************
     *******/
iR   */* extintui_lnk.c  v1.0
     */
1j   */* Jürgen Haage  Burgstraße 30
     */
VT   */* 6552 Bad Münster a. St. Ebernb
     urg 2 */
Bf   */*******************************
     *******/
BM3    XREF   _ExtintuiBase
d0     XDEF   _ShowWindow
Gv     XDEF   _ShowScreen
9Z     XDEF   _ActScreen
Vx     XDEF   _RemScreen
6z     XDEF   _ActWindow
SN     XDEF   _RemWindow
No     XDEF   _PutCircle
2T     XDEF   _PutEllipse
vy     XDEF   _PutLine
hb     XDEF   _PutPoint
3m     XDEF   _SetColor
Ai     XDEF   _Mode
RS     XDEF   _GetPosX
YV     XDEF   _GetPosY
Xg     XDEF   _PrintText
QU     XDEF   _MouseX
Wb     XDEF   _MouseY
eM     XDEF   _MenuHeader
zP     XDEF   _MenuItem
Ov     XDEF   _MenuSubItem
vF     XDEF   _KillMenu
n6     XDEF   _SetStringInfo
ki     XDEF   _SetPropInfo
ms     XDEF   _SetGadget
fM     XDEF   _GetStrinfo
e0     XDEF   _GetPropX
h4     XDEF   _GetPropY
ez     XDEF   _GetPropW
BH     XDEF   _GetPropH
ji     XDEF   _ChangeProp
fy     XDEF   _ShowRequester
qW     XDEF   _RequesterEnd
Ih     XDEF   _Message
MS     XDEF   _Class
J7     XDEF   _Code
IX     XDEF   _Gadgetid
tm     XDEF   _Menuid
Mw     XDEF   _Itemid
BZ     XDEF   _Subid
xR     XDEF   _SetPal
tV     CSEG
F30  _ShowWindow:
Yu3    move.l   a6,-(a7)
1X     movea.l  _ExtintuiBase,a6
vU     movem.l  8(a7),d0-d5
```

Listing 2.
C-Programmierer linken »extintui_lnk.asm« zu ihrem Programm

```
ME     move.l   32(a7),a0
hS     jsr   -36(a6)
LC     movea.l  (a7)+,a6
xy     rts
2G0 _ShowScreen:
g23    move.l   a6,-(a7)
tf     movea.l  _ExtintuiBase,a6
vQ     movem.l  8(a7),d0-d1
QU     move.l   16(a7),a0
ZH     jsr   -42(a6)
TK     movea.l  (a7)+,a6
56     rts
Ew0 _ActScreen:
oA3    move.l   a6,-(a7)
1n     movea.l  _ExtintuiBase,a6
HI     move.l   8(a7),a0
Ay     jsr   -48(a6)
aR     movea.l  (a7)+,a6
CD     rts
jd0 _RemScreen:
vH3    move.l   a6,-(a7)
8u     movea.l  _ExtintuiBase,a6
OP     move.l   8(a7),a0
1m     jsr   -54(a6)
hY     movea.l  (a7)+,a6
JK     rts
SV0 _ActWindow:
2O3    move.l   a6,-(a7)
F1     movea.l  _ExtintuiBase,a6
VW     move.l   8(a7),a0
sa     jsr   -60(a6)
of     movea.l  (a7)+,a6
QR     rts
xC0 _RemWindow:
9V3    move.l   a6,-(a7)
M8     movea.l  _ExtintuiBase,a6
od     move.l   8(a7),a0
TH     jsr   -66(a6)
vm     movea.l  (a7)+,a6
XY     rts
mo0 _PutCircle:
Gc3    move.l   a6,-(a7)
TF     movea.l  _ExtintuiBase,a6
X3     movem.l  8(a7),d0-d2
K5     jsr   -72(a6)
2t     movea.l  (a7)+,a6
ef     rts
bp0 _PutEllipse:
NJ3    move.l   a6,-(a7)
aM     movea.l  _ExtintuiBase,a6
gD     movem.l  8(a7),d0-d3
vm     jsr   -78(a6)
90     movea.l  (a7)+,a6
1m     rts
Gu0 _PutLine:
Uq3    move.l   a6,-(a7)
hT     movea.l  _ExtintuiBase,a6
nK     movem.l  8(a7),d0-d3
ma     jsr   -84(a6)
07     movea.l  (a7)+,a6
st     rts
UI0 _PutPoint:
bx3    move.l   a6,-(a7)
oa     movea.l  _ExtintuiBase,a6
qL     movem.l  8(a7),d0-d1
d0     jsr   -90(a6)
NE     movea.l  (a7)+,a6
z0     rts
Vy0 _SetColor:
143    move.l   a6,-(a7)
vn     movea.l  _ExtintuiBase,a6
zV     movem.l  8(a7),d0-d2
E5     jsr   -96(a6)
UL     movea.l  (a7)+,a6
67     rts
MR0 _Mode:
pB3    move.l   a6,-(a7)
2o     movea.l  _ExtintuiBase,a6
OS     move.l   8(a7),d0
7E     jsr   -102(a6)
bS     movea.l  (a7)+,a6
DE     rts
EB0 _GetPosX:
wI3    move.l   a6,-(a7)
9v     movea.l  _ExtintuiBase,a6
nW     jsr   -108(a6)
hY     movea.l  (a7)+,a6
JK     rts
OM0 _GetPosY:
203    move.l   a6,-(a7)
F1     movea.l  _ExtintuiBase,a6
aa     jsr   -114(a6)
ne     movea.l  (a7)+,a6
PQ     rts
BA0 _PrintText:
8U3    move.l   a6,-(a7)
L7     movea.l  _ExtintuiBase,a6
bc     move.l   8(a7),a0
ed     movem.l  12(a7),d0-d1
Pg     jsr   -120(a6)
vm     movea.l  (a7)+,a6
XY     rts
I10 _MouseX:
Gc3    move.l   a6,-(a7)
TF     movea.l  _ExtintuiBase,a6
5y     jsr   -126(a6)
1s     movea.l  (a7)+,a6
de     rts
Rv0 _MouseY:
M13    move.l   a6,-(a7)
ZL     movea.l  _ExtintuiBase,a6
s2     jsr   -132(a6)
7y     movea.l  (a7)+,a6
jk     rts
kZ0 _MenuHeader:
So3    move.l   a6,-(a7)
fR     movea.l  _ExtintuiBase,a6
vw     move.l   8(a7),a0
ZL     jsr   -138(a6)
E5     movea.l  (a7)+,a6
qr     rts
Xr0 _MenuItem:
Zv3    move.l   a6,-(a7)
mY     movea.l  _ExtintuiBase,a6
23     move.l   8(a7),a0
NQ     jsr   -144(a6)
LC     movea.l  (a7)+,a6
xy     rts
A40 _MenuSubItem:
g23    move.l   a6,-(a7)
tf     movea.l  _ExtintuiBase,a6
9A     move.l   8(a7),a0
BV     jsr   -150(a6)
SJ     movea.l  (a7)+,a6
45     rts
WW0 _KillMenu:
n93    move.l   a6,-(a7)
Om     movea.l  _ExtintuiBase,a6
GH     move.l   8(a7),a0
so     jsr   -156(a6)
ZQ     movea.l  (a7)+,a6
BC     rts
q20 _SetStringInfo:
uG3    move.l   a6,-(a7)
7t     movea.l  _ExtintuiBase,a6
NO     move.l   8(a7),a0
gt     jsr   -162(a6)
gX     movea.l  (a7)+,a6
IJ     rts
rP0 _SetPropInfo:
1N3    move.l   a6,-(a7)
E0     movea.l  _ExtintuiBase,a6
Mu     movem.l  8(a7),d0-d4
NC     jsr   -168(a6)
ne     movea.l  (a7)+,a6
PQ     rts
MY0 _SetGadget:
8U3    move.l   a6,-(a7)
L7     movea.l  _ExtintuiBase,a6
wY     movem.l  8(a7),d0-d5/a0-a2
BH     jsr   -174(a6)
ul     movea.l  (a7)+,a6
WX     rts
ov0 _GetStrinfo:
Fb3    move.l   a6,-(a7)
SE     movea.l  _ExtintuiBase,a6
z2     move.l   12(a7),d0
jk     move.l   8(a7),a0
ON     jsr   -180(a6)
2t     movea.l  (a7)+,a6
ef     rts
3B0 _GetPropX:
NJ3    move.l   a6,-(a7)
aM     movea.l  _ExtintuiBase,a6
7A     move.l   12(a7),d0
rs     move.l   8(a7),a0
ih     jsr   -186(a6)
A1     movea.l  (a7)+,a6
mn     rts
GP0 _GetPropY:
Vr3    move.l   a6,-(a7)
1U     movea.l  _ExtintuiBase,a6
FI     move.l   12(a7),d0
z0     move.l   8(a7),a0
Xn     jsr   -192(a6)
I9     movea.l  (a7)+,a6
uv     rts
EL0 _GetPropW:
dz3    move.l   a6,-(a7)
qc     movea.l  _ExtintuiBase,a6
NQ     move.l   12(a7),d0
78     move.l   8(a7),a0
F7     jsr   -198(a6)
QH     movea.l  (a7)+,a6
23     rts
910 _GetPropH:
173    move.l   a6,-(a7)
yk     movea.l  _ExtintuiBase,a6
VY     move.l   12(a7),d0
FG     move.l   8(a7),a0
KK     jsr   -204(a6)
YP     movea.l  (a7)+,a6
AB     rts
6p0 _ChangeProp:
tF3    move.l   a6,-(a7)
6s     movea.l  _ExtintuiBase,a6
MN     move.l   8(a7),a0
Ya     movem.l  12(a7),d0-d4
9Q     jsr   -210(a6)
gX     movea.l  (a7)+,a6
IJ     rts
DT0 _ShowRequester:
1N3    move.l   a6,-(a7)
E0     movea.l  _ExtintuiBase,a6
Mu     movem.l  8(a7),d0-d4
qJ     jsr   -216(a6)
ne     movea.l  (a7)+,a6
PQ     rts
d00 _RequesterEnd:
8U3    move.l   a6,-(a7)
L7     movea.l  _ExtintuiBase,a6
bc     move.l   8(a7),a0
eo     jsr   -222(a6)
ul     movea.l  (a7)+,a6
WX     rts
wV0 _Message:
Fb3    move.l   a6,-(a7)
SE     movea.l  _ExtintuiBase,a6
ij     move.l   8(a7),a0
L7     jsr   -228(a6)
1s     movea.l  (a7)+,a6
de     rts
Qg0 _Class:
Mi3    move.l   a6,-(a7)
ZL     movea.l  _ExtintuiBase,a6
8B     jsr   -234(a6)
7y     movea.l  (a7)+,a6
jk     rts
VQ0 _Code:
So3    move.l   a6,-(a7)
fR     movea.l  _ExtintuiBase,a6
vF     jsr   -240(a6)
D4     movea.l  (a7)+,a6
pq     rts
FW0 _Gadgetid:
Yu3    move.l   a6,-(a7)
1X     movea.l  _ExtintuiBase,a6
bX     jsr   -246(a6)
JA     movea.l  (a7)+,a6
vw     rts
Nh0 _Menuid:
e03    move.l   a6,-(a7)
```

```
rd       movea.l   _ExtintuiBase,a6
0b       jsr    -252(a6)
PG       movea.l   (a7)+,a6
12       rts
kc0 _Itemid:
k63      move.l    a6,-(a7)
xj       movea.l   _ExtintuiBase,a6
4t       jsr    -258(a6)
VM       movea.l   (a7)+,a6
78       rts
ty0 _Subid:
qC3      move.l    a6,-(a7)
3p       movea.l   _ExtintuiBase,a6
rx       jsr    -264(a6)
bS       movea.l   (a7)+,a6
DE       rts
zt0 _SetPal:
wI3      move.l    a6,-(a7)
9v       movea.l   _ExtintuiBase,a6
Fm       movem.l   8(a7),d0-d3
f2       jsr    -270(a6)
1Z       movea.l   (a7)+,a6
KL       rts
f8       end
(C) 1988 M&T
```

Listing 2. (Schluß)

Programmname:	extintui_lib.fd
Computer:	A500, A1000, A2000 mit Kickstart 1.2 und 1.3
Sprache:	ASCII-Text
Bemerkung:	Siehe Text

```
Programmautor: Jürgen Haage
---------------------------
rm0 * extintui_bas.fd, v1.0
VO  * Jürgen Haage, Burgstarße 30
Xe  * 6552 Bad Münster am Stein
br  ##base _extintuibase
%0  ##bias 30
32  ##public
Vx  FindLib()()
0t  ShowWindow(left,top,width,height,idcmp,flags,titel)(D0,D1,D2
    ,D3,D4,D5,A0)
eU  ShowScreen(depth,mode,titel)(D0,D1,A0)
PK  ActScreen(screen)(A0)
c9  RemScreen(screen)(A0)
sh  ActWindow(window)(A0)
56  RemWindow(window)(A0)
WL  PutCircle(x,y,radius)(D0,D1,D2)
Ux  PutEllipse(x,y,vradius,hradius)(D0,D1,D2,D3)
ma  PutLine(x1,y1,x2,y2)(D0,D1,D2,D3)
eN  PutPoint(x,y)(D0,D1)
Fr  SetColor(front,back,out)(D0,D1,D2)
C9  Mode(mode)(D0)
YT  GetPosX()()
cY  GetPosY()()
am  PrintText(text,x,y)(A0,D0,D1)
nm  MouseX()()
qq  MouseY()()
BE  MenuHeader(text)(A0)
gN  MenuItem(text)(A0)
cE  MenuSubItem(text)(A0)
aw  KillMenu(window)(A0)
Aa  SetStringInfo(buffer)(A0)
1B  SetPropInfo(flags,hpot,vpot,hbody,vbody)(D0,D1,D2,D3,D4)
mZ  SetGadget(left,top,width,height,active,type,text,info,id)(D0
    ,D1,D2,D3,D4,D5,A0,A1,A2)
Io  GetStrinfo(req,id)(A0,D0)
8h  GetPropX(req,id)(A0,D0)
Dn  GetPropY(req,id)(A0,D0)
6e  GetPropW(req,id)(A0,D0)
9S  GetPropH(req,id)(A0,D0)
KS  ChangeProp(req,id,x,y,w,h)(A0,D0,D1,D2,D3,D4)
tR  ShowRequester(left,top,width,height,color)(D0,D1,D2,D3,D4)
Ps  RequesterEnd(req)(A0)
PK  Message(window)(A0)
Kn  Class()()
cZ  Code()()
je  Gadgetid()()
8o  Menuid()()
W6  Itemid()()
Al  Subid()()
7P  SetPal(regnr,red,green,blue)(D0,D1,D2,D3)
LQ  ##end
(C) 1988 M&T
```

Listing 3. Basic-Programmierer generieren aus »extintui_lib.fd« eine ».bmap«-Datei

Programmname:	Vorspann
Computer:	A500, A1000, A2000 mit Kickstart 1.2 und 1.3
Sprache:	Amiga-Basic 1.2
Bemerkung:	Siehe Text

```
Programmautor: Jürgen Haage
---------------------------
HS4      DECLARE FUNCTION ShowScreen&(Tiefe,Modus,Titel) LIBRARY
X50  REM DECLARE FUNCTION ActScreen(screen) LIBRARY
wg   REM DECLARE FUNCTION RemScreen(screen) LIBRARY
hL4      DECLARE FUNCTION ShowWindow&(LeftEdge,TopEdge,WIDTH,Heig
         ht,Idcmp,Flags,Titel) LIBRARY
VrO  REM DECLARE FUNCTION ActWindow(window) LIBRARY
uS   REM DECLARE FUNCTION RemWindow(window) LIBRARY
Cr   REM DECLARE FUNCTION PutCircle(x,y,radius) LIBRARY
wO   REM DECLARE FUNCTION PutEllipse(x,y,vradius,hradius) LIBRARY
dM   REM DECLARE FUNCTION PutLine (x1,y1,x2,y2) LIBRARY
3S   REM DECLARE FUNCTION PutPoint(x,y) LIBRARY
Uf   REM DECLARE FUNCTION PrintText(Text,x,y) LIBRARY
WA   REM DECLARE FUNCTION SetColor(front,back,out) LIBRARY
81   REM DECLARE FUNCTION Mode(DrawMode) LIBRARY
z44      DECLARE FUNCTION GetPosX%() LIBRARY
5B       DECLARE FUNCTION GetPosY%() LIBRARY
kO       DECLARE FUNCTION MouseX%() LIBRARY
p6       DECLARE FUNCTION MouseY%() LIBRARY
4z       DECLARE FUNCTION MenuHeader%(Textadresse) LIBRARY
UN       DECLARE FUNCTION MenuItem%(Textadresse) LIBRARY
PI       DECLARE FUNCTION MenuSubItem%(Textadresse) LIBRARY
RzO  REM DECLARE FUNCTION KillMenu(window) LIBRARY
LW4      DECLARE FUNCTION SetStringInfo&(maxchars) LIBRARY
a4       DECLARE FUNCTION SetPropInfo&(Flags,hpot,vpot,hbody,vbod
         y) LIBRARY
37       DECLARE FUNCTION SetGadget&(LeftEdge,TopEdge,WIDTH,Heigh
         t,Activ,Typ,Text,Info,Id) LIBRARY
MJ       DECLARE FUNCTION GetStrinfo&(requester,Id) LIBRARY
PO       DECLARE FUNCTION GetPropX%(requester,Id) LIBRARY
W3       DECLARE FUNCTION GetPropY%(requester,Id) LIBRARY
LO       DECLARE FUNCTION GetPropW%(requester,Id) LIBRARY
uX       DECLARE FUNCTION GetPropH%(requester,Id) LIBRARY
2EO  REM DECLARE FUNCTION ChangeProp(requester,id,hpot,vpot,hbody
     ,vbody) LIBRARY
Jv4      DECLARE FUNCTION ShowRequester&(LeftEdge,TopEdge,WIDTH,H
         eight,Farbe) LIBRARY
FS       DECLARE FUNCTION RequesterEnd%(requester) LIBRARY
Dk       DECLARE FUNCTION Message%(WINDOW) LIBRARY
hz       DECLARE FUNCTION Class%() LIBRARY
6m       DECLARE FUNCTION Code%() LIBRARY
mf       DECLARE FUNCTION Gadgetid%() LIBRARY
dT       DECLARE FUNCTION Menuid%() LIBRARY
Tw       DECLARE FUNCTION Itemid%() LIBRARY
ao       DECLARE FUNCTION Subid%() LIBRARY
GnO  REM DECLARE FUNCTION SetPal(RegNr,red,green,blue)
gi   LIBRARY "extintui.library"
6C   ON ERROR GOTO CloseAll
QT   ON BREAK GOSUB CloseAll
gt   BREAK ON
zX   REM ***** Hier das Hauptprogramm *****
LG1   GOTO CloseAll
VQO  END
n9   CloseAll:
WJ2    IF ERR=150 THEN CLS:PRINT "Fehler bei Bibliotheksaufruf":F
       OR r=0 TO 2000:NEXT r
Sg     LIBRARY CLOSE
L2     BREAK OFF
ux     ON ERROR GOTO 0
bWO  END
(C) 1988 M&T
```

Listing 4. In dieses Listing bauen Sie ein Basic-Programm ein

Programmname:	extintui__fcn.c
Computer:	A500, A1000, A2000 mit Kickstart 1.2 und 1.3
Sprache:	C
Compiler:	Aztec-C V3.4, V3.6
Aufrufe:	cc extintui__fcn +L +B -S ln -o extintui.library extintui__lib.o extintui__fcn.o -LCL32
Bemerkung:	Funktionen der extintui.library

Programmautor: Jürgen Haage

```
/*************************************************/
/* extintui_fcn.c v1.0                           */
/*                                               */
/* Jürgen Haage  Burgstraße 30                   */
/* 6552 Bad Münster a. St. Ebernburg 2           */
/*************************************************/
#include "intuition/intuition.h"
#include "graphics/gfxmacros.h"
#include "exec/memory.h"
struct extintlib
  {
   struct Screen       *actscreen;
   struct Window       *actwindow;
   struct RastPort     *actrp;
   struct Gadget       *gadget;
   struct IntuiMessage message;
  };
FindLib()
{
struct Task *current;
while(!(current = (struct Task *)FindTask(0)));
return(current->tc_UserData);
}
ShowWindow(x,y,width,height,idcmp,flags,str)
SHORT x,y,width,height;
ULONG idcmp,flags;
BYTE *str;
{
struct NewWindow newwindow;
struct Window *window;
struct Screen *screen;
long *test;
struct extintlib *user;
if(!(user = (struct extintlib *)FindLib()))
  return(0L);
if(user->actscreen)
  {
  if((width + x) > user->actscreen->Width)
    width = user->actscreen->Width - x;
  if((height + y) > user->actscreen->Height)
    height = user->actscreen->Height - y;
  }
else
  {
  if((width + x) > 640)
    width = 640 - x;
  if((height + y) > 256)
    height = 256 - y;
  }
newwindow.LeftEdge = x;
newwindow.TopEdge = y;
newwindow.Width = width;
newwindow.Height = height;
newwindow.DetailPen = 0;
newwindow.BlockPen = 1;
newwindow.CheckMark = 0L;
newwindow.MinWidth = 10;
newwindow.MinHeight = 10;
newwindow.MaxWidth = 640;
newwindow.MaxHeight = 256;
newwindow.FirstGadget = 0L;
newwindow.IDCMPFlags = idcmp;
newwindow.Flags = flags;
newwindow.Title = str;
newwindow.Type = WBENCHSCREEN;
if(user->actscreen)
  {
  newwindow.Screen = (struct Screen *)user->actscreen;
  newwindow.Type = CUSTOMSCREEN;
  }
if(window = (struct Window *)OpenWindow(&newwindow))
  {
  user->actwindow = (struct Window *)window;
  user->actrp     = (struct RastPort *)window->RPort;
  return(window);
  }
return(0L);
}
ShowScreen(depth,mod,tit)
SHORT depth;
USHORT mod;
BYTE *tit;
{
struct extintlib *user;
struct NewScreen *newscreen;
struct Screen *screen;
if((depth > 4 && (mod == 1 || mod == 3)) || (depth > 5 &&
 mod == 2) || (depth > 6) || (mod > 3))
  return(0L);
if(!(newscreen = (struct NewScreen *)AllocMem(sizeof(struct
 NewScreen),MEMF_CLEAR)))
  return(0L);
newscreen->LeftEdge = 0;
newscreen->TopEdge = 0;
newscreen->Depth = depth;
newscreen->Width = 320;
newscreen->Height = 256;
newscreen->DetailPen = 0;
newscreen->BlockPen = 1;
newscreen->ViewModes = 0;
if(mod > 0 && mod != 2)
  {
  newscreen->Width = 640;
  newscreen->ViewModes = HIRES;
  }
if(mod > 1)
  {
  newscreen->Height = 512;
  newscreen->ViewModes |= LACE;
  }
if(mod==0 && depth==6)
  newscreen->ViewModes |= EXTRA_HALFBRITE;
newscreen->Type = CUSTOMSCREEN;          /* Customscreen */
newscreen->DefaultTitle = tit;
if(!(user = (struct extintlib *)FindLib()))
  return(0L);
if(screen = (struct Screen *)OpenScreen(newscreen))
  {
  FreeMem(newscreen,sizeof(struct NewScreen));
  user->actscreen = (struct Screen *)screen;
  return(screen);
  }
FreeMem(newscreen,sizeof(struct NewScreen));
return(0L);
}
ActScreen(screen)
struct Screen *screen;
{
struct extintlib *user;
if(!screen)
  return;
if(!(user = (struct extintlib *)FindLib()))
  return;
ScreenToFront(screen);
user->actscreen = (struct Screen *)screen;
}
RemScreen(screen)
struct Screen *screen;
{
struct Window *remwindow,*copywindow;
struct extintlib *user;
struct Gadget *gadget,*gadget2;
if(!screen)
  return;
if(remwindow = (struct Window *)screen->FirstWindow)
  {
  do
    {
    copywindow = (struct Window *)remwindow->NextWindow;
    CloseWindow(remwindow);
    remwindow = copywindow;
```

```
Iw       } while(remwindow);
Rw2    }
gv0  CloseScreen(screen);
Nz   if(!(user = (struct extintlib *)FindLib()))
V42    return;
LB0  user->actscreen = 0L;
MX   user->actwindow = 0L;
UL   user->actrp = 0L;
Y3   }
EH   ActWindow(window)
r7   struct Window *window;
X0   {
if   struct extintlib *user;
zA   if(!window)
fE2    return;
ZB0  if(!(user = (struct extintlib *)FindLib()))
hG2    return;
vY0  ActivateWindow(window);
nT   user->actwindow = (struct Window *)window;
lF   user->actrp = (struct RastPort *)window->RPort;
kF   }
cr   RemWindow(window)
3J   struct Window *window;
jC   {
ur   struct extintlib *user;
uD   struct Gadget *gadget,*gadget2;
CN   if(!window)
sR2    return;
m00  if(!(user = (struct extintlib *)FindLib()))
uT2    return;
li0  if(window == user->actwindow)
rK2    {
mx     user->actwindow = 0L;
ul     user->actrp = 0L;
yT     }
wT0  CloseWindow(window);
OV   }
ST   PutCircle(x,y,radius)
MF   SHORT x,y,radius;
zS   {
A7   struct extintlib *user;
zb   if(!(user = (struct extintlib *)FindLib()))
7g2    return;
qh0  if(user->actrp)
fC2    DrawEllipse(user->actrp,x,y,radius,radius); /* window ra
       stport */
9e0  }
lL   PutEllipse(x,y,vradius,hradius)
qU   SHORT x,y,vradius,hradius;
8b   {
JG   struct extintlib *user;
8k   if(!(user = (struct extintlib *)FindLib()))
Gp2    return;
zq0  if(user->actrp)
JN2    DrawEllipse(user->actrp,x,y,vradius,hradius); /* window
       rastport */
In0  }
cz   PutLine(x1,y1,x2,y2)
qE   SHORT x1,y1,x2,y2;
Hk   {
SP   struct extintlib *user;
Ht   if(!(user = (struct extintlib *)FindLib()))
Py2    return;
8z0  if(user->actrp)
Mp2    {
eJ     Move(user->actrp,x1,y1);
kI     Draw(user->actrp,x2,y2);
Ty     }
Uz0  }
5H   PutPoint(x,y)
10   SHORT x,y;
Tw   {
eb   struct extintlib *user;
T5   if(!(user = (struct extintlib *)FindLib()))
bA2    return;
KB0  if(user->actrp)
Y12    {
2w     Move(user->actrp,x,y);
01     Draw(user->actrp,x,y);
fA     }
gB0  }
b0   SetColor(front,back,out)
Ad   BYTE front,back,out;
f8   {
qn   struct extintlib *user;
fH   if(!(user = (struct extintlib *)FindLib()))
nM2    return;
WN0  if(user->actrp)
kD2    {
9k     SetAPen(user->actrp,front);
I1     SetBPen(user->actrp,back);
OT     SetOPen(user->actrp,out);
sN     }
t00  }
2s   Mode(mode)
zs   BYTE mode;
sL   {
30   struct extintlib *user;
sU   if(!(user = (struct extintlib *)FindLib()))
0Z2    return;
ja0  if(user->actrp)
Hc2    SetDrMd(user->actrp,mode);
2X0  }
DV   GetPosX()
OT   {
B8   struct extintlib *user;
0c   if(!(user = (struct extintlib *)FindLib()))
8h2    return;
ri0  if(user->actrp)
lv2    return(user->actrp->cp_x);
Af0  }
0h   GetPosY()
8b   {
JG   struct extintlib *user;
8k   if(!(user = (struct extintlib *)FindLib()))
Gp2    return;
zq0  if(user->actrp)
y92    return(user->actrp->cp_y);
In0  }
kJ   PrintText(text,x,y)
SQ   BYTE *text;
qp   SHORT x,y;
I1   {
TQ   struct extintlib *user;
Iu   if(!(user = (struct extintlib *)FindLib()))
Qz2    return;
900  if(user->actrp)
Nq2    {
rl     Move(user->actrp,x,y);
vX     Text(user->actrp,text,strlen(text));
Uz     }
V00  }
Qr   MouseX()
Tw   {
eb   struct extintlib *user;
T5   if(!(user = (struct extintlib *)FindLib()))
bA2    return;
7P0  if(user->actwindow)
yZ2    return(user->actwindow->MouseX);
d80  }
a2   MouseY()
b4   {
mj   struct extintlib *user;
bD   if(!(user = (struct extintlib *)FindLib()))
jI2    return;
FX0  if(user->actwindow)
Cj2    return(user->actwindow->MouseY);
1G0  }
sa   MenuHeader(text)
vt   BYTE *text;
kD   {
dk   SHORT maxw = 0, l = 0;
wt   struct extintlib *user;
XR   struct Menu *menu,*newmenu,*copymenu;
m0   if(!(user = (struct extintlib *)FindLib()))
MH2    return(0);
lq0  if(!(user->actwindow))
OJ2    return(0);
qn0  if(!(newmenu = (struct Menu *)AllocMem(sizeof(struct Menu),
     MEMF_CLEAR)))
QL2    return(0);
2P0  Forbid();
w0   maxw = (strlen(text)*8)+8;
iZ   if(menu = (struct Menu *)user->actwindow->MenuStrip)
xQ2    {
```

Listing 5. Die Funktionen für die »extintui.library«. Bitte mit dem Checksummer (Seite 143) eingeben.

```
3Q     do
zS4      {
es       copymenu = (struct Menu *)menu;
Bd       l = menu->LeftEdge;
9z       l += menu->Width;
8b       menu = (struct Menu *)menu->NextMenu;
C1       } while(menu);
ay2    copymenu->NextMenu = (struct Menu *)newmenu;
Af     }
OnO  else
xE2    user->actwindow->MenuStrip = (struct Menu *)newmenu;
A40  newmenu->LeftEdge  = 1;
mZ   newmenu->Width     = maxw;
T3   newmenu->Height    = 10;
QS   newmenu->Flags     = 1;
bG   newmenu->MenuName  = text;
FT   newmenu->FirstItem = 0L;
Rg   newmenu->NextMenu  = 0L;
kS   SetMenuStrip(user->actwindow,user->actwindow->MenuStrip)
     ;
eS   Permit();
tp   return(1);
Ns   }
Sj   MenuItem(text)
XV   BYTE *text;
Mp   {
XU   struct extintlib *user;
Co   struct Menu *menu;
av   struct MenuItem *menuitem,*newmenuitem,*copymenuitem;
n8   struct IntuiText *itext;
up   SHORT  top = 0,width;
Q2   if(!(user = (struct extintlib *)FindLib()))
Y72    return;
PU0  if(!(user->actwindow))
2x2    return(0);
s00  if(menu = (struct MenuItem *)user->actwindow->MenuStrip)
X02    {
RC     while(menu->NextMenu)
e74      menu = (struct Menu *)menu->NextMenu;
e92    }
UH0  else
942    return(0);
VJ0  if(!(newmenuitem = (struct MenuItem *)AllocMem(sizeof(struc
     t MenuItem),MEMF_CLEAR)))
B62    return(0);
Ld0  if(!(itext = (struct IntuiText **)AllocMem( sizeof( struct
     IntuiText),MEMF_CLEAR)))
g92    {
zc     FreeMem(newmenuitem,sizeof(struct MenuItem));
FA     return(0);
nI     }
390  itext->IText = (BYTE *)text;
Jq   itext->DrawMode = 2;
mI   width = IntuiTextLength(itext);
vI   Forbid();
s8   if(menuitem = (struct MenuItem *)menu->FirstItem)
pI2    {
vI     do
rK4      {
nf       copymenuitem = (struct MenuItem *)menuitem;
eI       if(width < menuitem->Width)
9f6        width = menuitem->Width;
ob4      else
k56        menuitem->Width = width;
1d4      top  = menuitem->TopEdge;
ZQ       menuitem = (struct MenuItem *)menuitem->NextItem;
BK       } while(menuitem);
1E2    top += 10;
wz     copymenuitem->NextItem = (struct MenuItem *)newmenuitem;
6b     }
wJ0  else
6Q2    menu->FirstItem = (struct MenuItem *)newmenuitem;
9C0  newmenuitem->LeftEdge = 0;
WC   newmenuitem->TopEdge  = top;
YS   newmenuitem->Width    = width;
q9   newmenuitem->Height   = 10;
Cc   newmenuitem->Flags    = (ITEMTEXT|ITEMENABLED|HIGHCOMP);
im   newmenuitem->ItemFill = (struct IntuiText *)itext;
S7   newmenuitem->NextItem = 0L;
LP   newmenuitem->SelectFill = 0L;
BH   newmenuitem->SubItem = 0L;
gA   newmenuitem->NextSelect = 0xFFFF;
JR   SetMenuStrip(user->actwindow,user->actwindow->MenuStrip)
     ;
dR   Permit();
so   return(1);
Mr   }
pC   MenuSubItem( text)
WU   BYTE *text;
Lo   {
ZM   struct extintlib       *user;
Bn   struct Menu      *menu;
kl   struct MenuItem  *menuitem, *copymenuitem, *submenuitem, *n
     ewsubitem;
m7   struct IntuiText *itext;
lf   SHORT            top = 0, width;
P1   if(!(user = (struct extintlib *)FindLib()))
X62    return;
OTO  if(!(user->actwindow))
1w2    return(0);
WA0  if( menu = ( struct Menu *)user->actwindow->MenuStrip)
W22    {
Xx     while( menu->NextMenu)
Wk4      menu = ( struct Menu *)menu->NextMenu;
d82    }
TG0  else
832    return(0);
MY0  if( menuitem = ( struct MenuItem *)menu->FirstItem)
d62    {
uS     while( menuitem->NextItem)
sL4      menuitem = ( struct MenuItem *)menuitem->NextItem;
kF2    }
aN0  else
FA2    return(0);
Ag0  if( !( newsubitem = ( struct MenuItem *)AllocMem( sizeof( s
     truct MenuItem),MEMF_CLEAR)))
HC2    return(0);
ZS0  if( !( itext = ( struct IntuiText *)AllocMem( sizeof( struc
     t IntuiText),MEMF_CLEAR)))
mF2    {
O0     FreeMem( newsubitem,sizeof( struct MenuItem));
LG     return(0);
t0     }
UA0  itext->IText = ( BYTE *)text;
pw   itext->DrawMode = 2;
Gc   width = IntuiTextLength( itext);
10   Forbid();
Gx   if( submenuitem = ( struct MenuItem *)menuitem->SubItem)
v02    {
10     do
xQ4      {
kU       copymenuitem = ( struct MenuItem *)submenuitem;
ZM       if( width < submenuitem->Width)
bN6        width = submenuitem->Width;
uh4      else
FE6        submenuitem->Width = width;
g54      top  = submenuitem->TopEdge;
3m       submenuitem = ( struct MenuItem *)submenuitem->NextIte
         m;
I9       } while( submenuitem);
rK2    top += 10;
jX     copymenuitem->NextItem = ( struct MenuItem *)newsubitem;
Cn     }
2p0  else
hR2    menuitem->SubItem = ( struct MenuItem *)newsubitem;
cM0  newsubitem->LeftEdge = menuitem->Width - 5;
Qq   newsubitem->TopEdge  = top;
Ps   newsubitem->Width    = width;
FA   newsubitem->Height   = 10;
JX   newsubitem->Flags    = ( ITEMTEXT|ITEMENABLED|HIGHCOMP);
SE   newsubitem->ItemFill = ( struct IntuiText *)itext;
hm   newsubitem->NextItem = 0L;
nf   newsubitem->SelectFill = 0L;
RV   newsubitem->SubItem = 0L;
I0   newsubitem->NextSelect = 0xFFFF;
pX   SetMenuStrip(user->actwindow,user->actwindow->MenuStrip)
     ;
JX   Permit();
yu   return(1);
Sx   }
eK   KillMenu( window)
11   struct Window *window;
Ru   {
04   struct Menu      *menu,*copymenu;
k9   struct MenuItem  *menuitem,*copymenuitem,*subitem,*copysubi
     tem;
u5   if(!window)
a92    return;
```

```
e10 Forbid();
3g  if( menu = ( struct Menu *)window->MenuStrip)
Y12  {
e1   do
a34   {
LX    if( menuitem = ( struct MenuItem *)menu->FirstItem)
c56    {
15     do
e73   {
g88       if( subitem = ( struct MenuItem *)menuitem->SubIte
          m)
g95     {
m9      do
iB7      {
7wC           copysubitem = ( struct MenuItem *)subitem->Nex
              tItem;
3y            FreeMem( subitem->ItemFill,sizeof( struct Intu
              iText));
w7            FreeMem( subitem,sizeof( struct MenuItem));
4h            subitem = ( struct MenuItem *)copysubitem;
RF7      } while( subitem);
sN5     }
GR8       copymenuitem = ( struct MenuItem *)menuitem->NextI
          tem;
DR        FreeMem( menuitem->ItemFill,sizeof( struct IntuiTe
          xt));
Mf        FreeMem( menuitem,sizeof( struct MenuItem));
tX        menuitem = ( struct MenuItem *)copymenuitem;
Ph3   } while( menuitem);
yT6    }
II4   copymenu = ( struct Menu *)menu->NextMenu;
JL    FreeMem( menu,sizeof( struct Menu));
Bs    menu = ( struct Menu *)copymenu;
66    } while( menu);
3Y2  }
pt0 window->MenuStrip = 0L;
0C  Permit();
6b  }
8D  SetStringInfo(buffer)
qR  SHORT buffer;
5Y  {
24  struct StringInfo *strinfo;
IA  if(strinfo = (struct StringInfo *)AllocMem(sizeof(struct St
    ringInfo),MEMF_CLEAR))
8b2  {
78   if(strinfo->Buffer = (BYTE *)AllocMem(((buffer+1)*2),MEM
     F_CLEAR))
Ad4   {
xg    strinfo->UndoBuffer = (BYTE *)(strinfo->Buffer)+buffe
      r+1;
Ae    strinfo->MaxChars = buffer;
SN    strinfo->DispPos  = 0;
aJ    strinfo->BufferPos = 0;
c0    return((struct StringInfo *)strinfo);
Kp    }
Na2  FreeMem(strinfo,sizeof(struct StringInfo));
Mr   }
ns0 return(0L);
0t  }
3W  SetPropInfo(flags,hbody,vbody,hpot,vpot)
bS  USHORT flags,hbody,vbody,hpot,vpot;
Nq  {
zW  struct PropInfo *propinfo;
m3  if(propinfo = (struct PropInfo *)AllocMem(sizeof(struct Pro
    pInfo),MEMF_CLEAR))
Qt2  {
B8   flags              |= AUTOKNOB;
P2   propinfo->Flags     = flags;
8A   propinfo->HorizPot  = hpot;
f6   propinfo->VertPot   = vpot;
xt   propinfo->HorizBody = hbody;
Ps   propinfo->VertBody  = vbody;
DF   return((struct PropInfo *)propinfo);
c7   }
380 return(0L);
e9  }
zq  SetGadget(x,y,weite,hohe,flag,typ,text,info,id)
Mg  SHORT x,y,weite,hohe;
Nc  USHORT typ,flag,id;
95  LONG info;
rp  BYTE *text;
g9  {
0d  struct Gadget *newgadget,*gadget,*copygadget;
5Q  struct IntuiText *itext;
tq  struct extintlib *user;
XR  struct Border *border;
js  SHORT err = 0,*array;
AT  USHORT fl;
1N  if(!(user = (struct extintlib *)FindLib()))
tS2   return;
Lz0 if(!user->actwindow)
KP2   return(0L);
7L0 if(newgadget = (struct Gadget *)AllocMem(sizeof(struct Gadg
    et),MEMF_CLEAR))
sL2  {
Ku   if(border = (struct Border *)AllocMem(sizeof(struct Borde
     r),MEMF_CLEAR))
uN4   {
aZ    if(array = (SHORT *)AllocMem(24,MEMF_CLEAR))
wP6    {
Hq     if(itext = (struct IntuiText *)AllocMem(sizeof(struct
        IntuiText),MEMF_CLEAR))
yR8     {
Vq      newgadget->GadgetRender = (struct Border *)border;
Gm      newgadget->GadgetText = (struct IntuiText *)itext;
LY3   border->XY = (SHORT *)array;
6b8     }
wJ6    else
za8     err = 1;
9e6    }
zm4   else
2d6    err = 1;
Ch4   }
2p2  else
5g4   err = 1;
Fk2  }
6p0 if(err)
Dg2  {
B9   if(newgadget)
cI4    FreeMem(newgadget,sizeof(struct Gadget));
1A2  if(array)
cr4    FreeMem(array,24);
S82  if(border)
2K4    FreeMem(border,sizeof(struct Border));
rm2  return(0);
Pu   }
Ur0 Forbid();
pk  if(gadget = (struct Gadget *)user->gadget)
0r2  {
Ur   do
Qt4   {
mE    copygadget = (struct Gadget *)gadget;
cv    gadget = (struct Gadget *)gadget->NextGadget;
wH    } while(gadget);
A52  copygadget->NextGadget = (struct Gadget *)newgadget;
Z4   }
PC0 else
Kq2  user->gadget = (struct Gadget *)newgadget;
sZ0 *(array)   = (SHORT)-1;
Gg  *(array+1) = (SHORT)-1;
Mn  *(array+2) = (SHORT)-1;
Rw  *(array+3) = (SHORT)hohe;
kK  *(array+4) = (SHORT)weite;
dA  *(array+5) = (SHORT)hohe;
wY  *(array+6) = (SHORT)weite;
qM  *(array+7) = (SHORT)-1;
wT  *(array+8) = (SHORT)-1;
2a  *(array+9) = (SHORT)-1;
GH  border->Count      = 5;
ut  border->DrawMode   = 2;
AJ  border->NextBorder = 0L;
gn  typ               |= REQGADGET;
Yq  itext->DrawMode    = 2;
hu  itext->IText       = text;
6s  if(typ == (REQGADGET| STRGADGET))
i42  itext->LeftEdge   = (IntuiTextLength(itext) + 5) * -1;
jW0 else
eq2  itext->LeftEdge   = (weite - IntuiTextLength(itext)) / 2
     ;
3x0 if(typ == (REQGADGET| PROPGADGET))
sm2  itext->TopEdge    = -11;
na0 else
s12  itext->TopEdge      = (hohe - 5) / 2;
3Z0 itext->NextText    = 0L;
cg  itext->ITextFont   = 0L;
NG  newgadget->LeftEdge = x;
```

Listing 5. (Fortsetzung)

```
Cz  newgadget->TopEdge  = y;
u4  newgadget->Width    = weite;
d0  newgadget->Height   = hohe;
Hj  flag               &= ~ENDGADGET;
GC  newgadget->Flags    = GADGHCOMP;
q7  newgadget->Activation = flag;
Qb  newgadget->GadgetType = typ;
R1  newgadget->GadgetID  = id;
h2  newgadget->NextGadget = 0L;
Vq  newgadget->SpecialInfo = 0L;
xh  if((typ == (REQGADGET|STRGADGET)) && info)
2n2   newgadget->SpecialInfo = (struct StringInfo *)info;
4r0 else
4N2   if((typ == (REQGADGET|PROPGADGET)) && info)
3V4     newgadget->SpecialInfo = (struct PropInfo *)info;
7u2   else
qz4     newgadget->GadgetType = (REQGADGET|BOOLGADGET);
dR0 Permit();
ZV  return(newgadget);
Mr  }
1g  GetStrinfo(requester,id)
VH  struct Requester *requester;
BZ  SHORT id;
Mp  {
7Z  struct Gadget *gadget;
1f  if(!requester)
ty2   return(0L);
J60 else
g72   if(!(gadget = (struct Gadget *)requester->ReqGadget))
w14     return(0L);
0J0 while(gadget->GadgetID != id && gadget)
ex2   gadget = (struct Gadget *)gadget->NextGadget;
MU0 if(gadget->GadgetType & STRGADGET)
XL2   return(gadget->SpecialInfo->Buffer);
QD0 else
272   return(0L);
d80 }
JG  GetPropX(requester,id)
mY  struct Requester *requester;
Sq  SHORT id;
d6  {
Oq  struct Gadget *gadget;
Iw  if(!requester)
D82   return(0);
aN0 else
x02   if(!(gadget = (struct Gadget *)requester->ReqGadget))
GB4     return(0);
fa0 while(gadget->GadgetID != id && gadget)
vE2   gadget = (struct Gadget *)gadget->NextGadget;
S30 if(gadget->GadgetType == (PROPGADGET|REQGADGET))
ex2   return(gadget->SpecialInfo->HorizPot);
hU0 else
MH2   return(0);
uP0 }
ec  GetPropY(requester,id)
3p  struct Requester *requester;
j7  SHORT id;
uN  {
f7  struct Gadget *gadget;
ZD  if(!requester)
UP2   return(0);
re0 else
Ef2   if(!(gadget = (struct Gadget *)requester->ReqGadget))
XS4     return(0);
wr0 while(gadget->GadgetID != id && gadget)
CV2   gadget = (struct Gadget *)gadget->NextGadget;
jK0 if(gadget->GadgetType == (PROPGADGET|REQGADGET))
b62   return(gadget->SpecialInfo->VertPot);
y10 else
dY2   return(0);
Bg0 }
nJ  GetPropW(requester,id)
K6  struct Requester *requester;
00  SHORT id;
Be  {
w0  struct Gadget *gadget;
qU  if(!requester)
1g2   return(0);
8v0 else
Vw2   if(!(gadget = (struct Gadget *)requester->ReqGadget))
oj4     return(0);
D80 while(gadget->GadgetID != id && gadget)
Tm2   gadget = (struct Gadget *)gadget->NextGadget;
0b0 if(gadget->GadgetType == (PROPGADGET|REQGADGET))
Rh2   return(gadget->SpecialInfo->HorizBody);
F20 else
up2   return(0);
Sx0 }
6n  GetPropH(requester,id)
bN  struct Requester *requester;
Hf  SHORT id;
Sv  {
Df  struct Gadget *gadget;
71  if(!requester)
2x2   return(0);
PC0 else
mD2   if(!(gadget = (struct Gadget *)requester->ReqGadget))
504     return(0);
UP0 while(gadget->GadgetID != id && gadget)
k32   gadget = (struct Gadget *)gadget->NextGadget;
Hs0 if(gadget->GadgetType == (PROPGADGET|REQGADGET))
9t2   return(gadget->SpecialInfo->VertBody);
WJ0 else
B62   return(0);
jE0 }
Sz  ChangeProp(requester,id,x,y,w,h)
se  struct Requester *requester;
Yw  SHORT id;
ug  USHORT x,y,w,h;
kD  {
Vx  struct Gadget *gadget;
x1  struct PropInfo *prop;
Q4  if(!requester)
tS2   return;
iV0 else
5W2   if(!(gadget = (struct Gadget *)requester->ReqGadget))
wV4     return;
ni0 while(gadget->GadgetID != id && gadget)
3M2   gadget = (struct Gadget *)gadget->NextGadget;
aB0 if(gadget->GadgetType == (PROPGADGET|REQGADGET))
v02   {
2h    prop = (struct PropInfo *)gadget->SpecialInfo;
wT    prop->HorizPot = x;
9Q    prop->VertPot  = y;
ry    prop->HorizBody = w;
VS    prop->VertBody  = h;
ud    RefreshGList(gadget,requester->RWindow,requester,1);
6b    }
7c0 }
ET  ShowRequester(x,y,w,h,farbe)
Tf  SHORT x,y,w,h,farbe;
6Z  {
y0  struct extintlib    *user;
J5  struct Requester *requester;
7j  if(!(user = (struct extintlib *)FindLib()))
ej2   return(0L);
hL0 if(!user->actwindow)
gl2   return(0L);
SC0 if(!(requester = (struct Requester *)AllocMem(sizeof(struct
     Requester),MEMF_CLEAR)))
in2   return(0L);
4e0 requester->ReqGadget = user->gadget;
s7  requester->LeftEdge = x;
hq  requester->TopEdge  = y;
Ao  requester->Width    = w;
Ro  requester->Height   = h;
Im  requester->BackFill = farbe;
FQ  if(Request(requester,user->actwindow))
v12   return(requester);
G30 else
sx2   return(0L);
Ty0 }
yV  RequesterEnd(requester)
c0  struct Requester *requester;
Sv  {
cv  struct Gadget *gadget,*gadget2;
HB  struct Border *border;
NO  struct StrinfInfo *strinfo;
gd  struct extintlib *user;
XP  USHORT type;
W8  if(!(user = (struct extintlib *)FindLib()))
382   return(0L);
yZ0 if(requester)
Jz2   EndRequest(requester,requester->RWindow);
4z0 if(gadget = (struct Gadget *)user->gadget)
d62   {
J6    do
f84     {
```

```
    gadget2 = (struct Gadget *)gadget;
    gadget = (struct Gadget *)gadget2->NextGadget;
    border = (struct Border *)gadget2->GadgetRender;
    FreeMem(border->XY,24);
    FreeMem(border,sizeof(struct Border));
    FreeMem(gadget2->GadgetText,sizeof(struct IntuiText));
    type = gadget2->GadgetType;
    if((type == (REQGADGET|STRGADGET)) && gadget2->SpecialInfo)
      {
      strinfo = (struct StringInfo *)gadget2->SpecialInfo;
      FreeMem(strinfo->Buffer,((strinfo->MaxChars) * 2));
      FreeMem(strinfo,sizeof(struct StringInfo));
      }
    else
      if((type == (REQGADGET|PROPGADGET)) && gadget2->SpecialInfo)
        FreeMem(gadget2->SpecialInfo,sizeof(struct PropInfo));
    FreeMem(gadget2,sizeof(struct Gadget));
    }
    while(gadget);
  }
user->gadget = 0L;
}
Message(window)
struct Window *window;
{
struct extintlib *user;
struct IntuiMessage *nachricht;
if(!(user = (struct extintlib *)FindLib()))
  return(0);
if(!window)
  if(!(window = (struct Window *)user->actwindow))
    return(0);
nachricht = (struct IntuiMessage *)GetMsg(window->UserPort);
if(nachricht)
  {
  movmem(nachricht,&(user->message),sizeof(struct IntuiMessage));
  ReplyMsg(nachricht);
  return(1);
  }
return(0);
}
Class()
{
struct extintlib *user;
if(!(user = (struct extintlib *)FindLib()))
  return(0);
return(user->message.Class);
}
Code()
{
struct extintlib *user;
if(!(user = (struct extintlib *)FindLib()))
  return(0);
return(user->message.Code);
}
Gadgetid()
{
struct extintlib *user;
struct Gadget *gadget;
if(!(user = (struct extintlib *)FindLib()))
  return(0);
gadget = (struct Gadget *)user->message.IAddress;
return(gadget->GadgetID);
}
Menuid()
{
struct extintlib *user;
if(!(user = (struct extintlib *)FindLib()))
  return(0);
return((user->message.Code & 0x1F));
}
Itemid()
{
struct extintlib *user;
if(!(user = (struct extintlib *)FindLib()))
  return(0);
return((user->message.Code >> 5) & 0x003F);
}
Subid()
{
struct extintlib *user;
if(!(user = (struct extintlib *)FindLib()))
  return(0);
return((user->message.Code >> 11) & 0x001F);
}
SetPal(regnr,red,green,blue)
USHORT regnr,red,green,blue;
{
struct Screen *screen;
struct extintlib *user;
if(!(user = (struct extintlib *)FindLib()))
  return;
if(red<0 || red>15 || green<0 || green>15 || blue<0 || blue>15 ||regnr<0 || regnr>31)
  return;
if(screen=(struct Screen *)user->actscreen)
  SetRGB4(&screen->ViewPort,regnr,red,green,blue);
}
(C) 1989 M&T
```

Listing 5. (Schluß)

Programmname:	M2Extintui.def
Computer:	A500, A1000, A2000 mit Kickstart 1.2 und 1.3
Sprache:	Modula-2
Compiler:	M2Amiga
Bemerkung:	siehe Text

```
Programmautor: Andreas Görtler
------------------------------
DEFINITION MODULE M2Extintui["extintui.library",1];
(* by ag  11.12.88 3:00 *)
(* 1.3    = Includefiles.h für Version 1.3
   M2     = M2-Amiga Modula-II
   fcn.c = Die Implementation der Routinen (extintui_fcn.c)
   3 Kriterien: - TypenÜbereinstimmung mit 1.3(Include),M2,und fcn.c
                - Rückgabewerte
                - Spezifikation der Parameter: z.B. WindowPtr anstatt ADDRESS *)
FROM SYSTEM    IMPORT ADDRESS;
FROM Exec      IMPORT UByte;
FROM Graphics  IMPORT DrawModeSet;
FROM Intuition IMPORT ScreenPtr,WindowPtr,RequesterPtr,IDCMPFlagSet,PropInfoPtr,
                      PropInfoFlagSet,ActivationFlagSet,StringInfoPtr,GadgetPtr,
                      WindowFlagSet;
(*--------------- Die Screen- und Window-Prozeduren -------------------------*)
PROCEDURE ShowScreen(depth[0]:INTEGER;
                     viewModes[1]:CARDINAL;
                     defaultTitle[8]:ADDRESS):ScreenPtr; CODE -42;
  (* if fail then return NIL else return ScreenPtr *)
  (* erzeugt neuen Screen mit übergebenen Parametern *)
PROCEDURE ActScreen(screenptr[8]:ScreenPtr); CODE -48;
PROCEDURE RemScreen(screenptr[8]:ScreenPtr); CODE -54;
PROCEDURE ShowWindow(leftEdge[0]:INTEGER;
                     topEdge[1]:INTEGER;
                     width[2]:INTEGER;
                     height[3]:INTEGER;
                     idcmpFlags[4]:IDCMPFlagSet;
                     flags[5]:WindowFlagSet;
                     title[8]:ADDRESS):WindowPtr; CODE -36;
  (* if fail then return NIL else return WindowPtr *)
PROCEDURE ActWindow(windowptr[8]:WindowPtr); CODE -60;
PROCEDURE RemWindow(windowptr[8]:WindowPtr); CODE -66;
(*----------------- Die Zeichenprozeduren ---------------------------------*)
PROCEDURE PutCircle(x[0]:INTEGER;
                    y[1]:INTEGER;
                    radius[2]:INTEGER); CODE -72;
PROCEDURE PutEllipse(x[0]:INTEGER;
                     y[1]:INTEGER;
```

Listing 6. Das nötige Listing für Modula-2-Programmierer: »M2Extintui.def«

```
39 DZ                       vradius[2]:INTEGER;
40 tk                       hradius[3]:INTEGER); CODE -78;
41 9q2   (* Typen: s.o. ..radius bei allen SHORT *)
42 dy0 PROCEDURE PutLine(x1[0]:INTEGER;
43 eqI                   y1[1]:INTEGER;
44 ly                    x2[2]:INTEGER;
45 fZ                    y2[3]:INTEGER); CODE -84;
46 Cd0 PROCEDURE PutPoint(x[0]:INTEGER;
47 jqJ                    y[1]:INTEGER); CODE -90;
48 vx2   (* setzt RastPort auf (x1,y1) bzw (x,y) und zeichnet mit
       Graphics.Draw bis
49 Z75      (x2,y2) bzw (x,y).*)
50 OF0 PROCEDURE PrintText(textptr[8]:ADDRESS;
51 8GK                     x[0]:INTEGER;
52 HN                      y[1]:INTEGER); CODE -120;
53 PF0 PROCEDURE Mode(drawMode[0]:DrawModeSet); CODE -102;
54 GT2   (* drawMode liefert Eintrag für aktuellen RastPort; Graph
       ics.SetDrMd wird
55 Tv5      aufgerufen *)
56 1T2   (* drawMode in 1.3 und fcn.c vom Typ BYTE, in M2 DrawMode
       Set =
57 NO5      Bytegröße *)
58 d10 PROCEDURE SetColor(fgPen[0]:CARDINAL;
59 TEJ                    bgPen[1]:CARDINAL;
60 oy                     aOlPen[2]:CARDINAL); CODE -96;
61 re2   (* In M2 RastPort.fgPen,.bgPen,.aOlPen : UByte; Die Argum
       ente für Set_Pen :
62 2n5      CARDINAL; In 1.3 FgPen,BgPen,AOlPen : BYTE; *)
63 bm0 PROCEDURE SetPal(n[0]:CARDINAL;
64 7GH                  r[1]:CARDINAL;
65 qp                   g[2]:CARDINAL;
66 Cs                   b[3]:CARDINAL); CODE -270;
67 Qx2   (* entspricht Graphics.SetRGB4 *)
68 Jt    (* Parameter sind in 1.3 und M2 USHORT (siehe ColorTable)
       ; fcn.c : SHORT! *)
69 Uo0 PROCEDURE GetPosX():INTEGER; CODE -108;
70 Ld  PROCEDURE GetPosY():INTEGER; CODE -114;
71 Z12   (* liefert RastPort.x bzw .y *)
72 Pt    (* in M2 und 1.3 : SHORT *)
73 v30 PROCEDURE MouseX():INTEGER; CODE -126;
74 ou  PROCEDURE MouseY():INTEGER; CODE -132;
75 G32   (* liefert Window.mouseX bzw .mouseY *)
76 IX0 (*------------------- Die Menüleisten ---------------------
    -----------------*)
77 Bq  PROCEDURE MenuHeader(textptr[8]:ADDRESS):INTEGER; CODE -138
    ;
78 SV2   (* liefert bei Erfolg 1,sonst 0; Ergebnis könnte auch vom
        Typ CARDINAL sein *)
79 WC    (* textptr liefert Eintrag für Menu.menuName; bei allen v
       om Typ APTR *)
80 4f0 PROCEDURE MenuItem(textptr[8]:ADDRESS):INTEGER; CODE -144;
81 VY2   (* liefert bei Erfolg 1,sonst 0; Ergebnis könnte auch vom
        Typ CARDINAL sein *)
82 zt    (* textptr liefert Eintrag für IntuiText.iText; bei allen
        vom Typ APTR *)
83 Kk0 PROCEDURE MenuSubItem(textptr[8]:ADDRESS):INTEGER; CODE -15
    0;
84 Yb2   (* liefert bei Erfolg 1,sonst 0; Ergebnis könnte auch vom
        Typ CARDINAL sein *)
85 2w    (* textptr liefert Eintrag für IntuiText.iText; bei allen
        vom Typ APTR *)
86 dy0 PROCEDURE KillMenu(windowptr[8]:WindowPtr); CODE -156;
87 2d  (*-------------------- Gadgets und Requester --------------
    ------------------*)
88 pa  PROCEDURE SetStringInfo(maxChars[8]:ADDRESS):StringInfoPtr;
     CODE -162;
89 Sw2   (* if fail then return NIL else return StringInfoPtr *)
90 gd0 PROCEDURE SetPropInfo(flags[0]:PropInfoFlagSet;
91 COM                       horizPot[1]:CARDINAL;
92 Gd                        vertPot[2]:CARDINAL;
93 GL                        horizBody[3]:CARDINAL;
94 gn                        vertBody[4]:CARDINAL):PropInfoPtr; CO
                          DE -168;
95 Ws2   (* if fail then return NIL else return new PropInfo with
       APTR *)
  (* In M2 und 1.3 und fcn.c alle Parameter vom Typ USHORT
       *)
97 170 PROCEDURE SetGadget(leftEdge[0]:INTEGER;
98 jbK                     topEdge[1]:INTEGER;
99 y3                      width[2]:INTEGER;
100 37                     height[3]:INTEGER;
101 2Y                     activation[4]:ActivationFlagSet;
102 u7                     gadgetType[5]:CARDINAL;
103 Be                     gadgetText[8]:ADDRESS;
104 ol                     specialInfo[9]:ADDRESS;
105 XK                     gadgetID[10]:INTEGER):GadgetPtr; CODE -
                        174;
106 yL2   (* if actwindow = NIL then return NIL else  Gadgetadresse
        *)
107 vq    (* Alle Typen in fcn.c und 1.3 o.k.; in M2 ist gadgetID v
        om Typ SHORT *)
108 QL0 PROCEDURE GetStrinfo(requesterptr[8]:RequesterPtr;
109 wQL                      gadgetID[0]:INTEGER):ADDRESS; CODE -18
                          0;
110 Ag2   (* fail -> return NIL, sonst StringInfo.buffer *)
111 sM0 PROCEDURE GetPropH(requesterptr[8]:RequesterPtr;
112 4QJ                    gadgetID[0]:INTEGER):CARDINAL; CODE -204
                        ;
113 Dt2   (* fail -> return 0, sonst PropInfo.vertBody *)
114 D00 PROCEDURE GetPropW(req[8]:RequesterPtr;
115 AOJ                    gadgetID[0]:INTEGER):CARDINAL; CODE -198
                        ;
116 jK2   (* fail -> return 0, sonst PropInfo.horizBody *)
117 KW0 PROCEDURE GetPropX(req[8]:RequesterPtr;
118 wHJ                    gadgetID[0]:INTEGER):CARDINAL; CODE -186
                        ;
119 pe2   (* fail -> return 0, sonst PropInfo.horizPot *)
120 Re0 PROCEDURE GetPropY(req[8]:RequesterPtr;
121 gIJ                    gadgetID[0]:INTEGER):CARDINAL; CODE -192
                        ;
122 A52   (* fail -> return 0, sonst PropInfo.vertPot *)
123 pE0 PROCEDURE ChangeProp(requesterptr[8]:RequesterPtr;
124 tlL                      gadgetID[0]:INTEGER;
125 kY                       horizPot[1]:CARDINAL;
126 oB                       vertPot[2]:CARDINAL;
127 ot                       horizBody[3]:CARDINAL;
128 Dx                       vertBody[4]:CARDINAL); CODE -210;
129 uW2   (* gadgetID in M2 und fcn.c SHORT, in 1.3 dagegen USHORT
        ! *)
130 1Y    (* restlichen Par. in M2 und 1.3 und fcn.c USHORT *)
131 V00 PROCEDURE ShowRequester(leftEdge[0]:INTEGER;
132 H90                         topEdge[1]:INTEGER;
133 Wb                          width[2]:INTEGER;
134 bf                          height[3]:INTEGER;
135 I1                          backFill[4]:UByte):RequesterPtr; CO
                             DE -216;
136 A14     (* if fail then return NIL else return RequesterPtr *)
137 wZ      (* In 1.3 und M2 backFill : UBYTE; in fcn.c SHORT! *)
138 qE0 PROCEDURE RequesterEnd(requesterptr[8]:RequesterPtr); CODE
    -222;
139 YL  (*------------------------ Nachrichtenempfang ------------
    ---------------*)
140 RA  PROCEDURE Message(windowptr[8]:WindowPtr):INTEGER; CODE -22
    8;
141 TW2   (* liefert bei Erfolg 1,sonst 0; Ergebnis könnte auch vom
         Typ CARDINAL sein *)
142 XI0 PROCEDURE Class():IDCMPFlagSet; CODE -234;
143 JF  PROCEDURE Code():CARDINAL; CODE -240;
144 Ge2   (* fail -> return 0, sonst IntuiMessage.class bzw. .code
        *)
145 i00 PROCEDURE Gadgetid():INTEGER; CODE -246;
146 Fk2   (* fail -> return 0, sonst Gadget.gadgetID *)
147 Lw    (* im M2 INTEGER, in 1.3 aber USHORT, siehe Hinweis unten
        ! *)
148 qy0 PROCEDURE Menuid():CARDINAL; CODE -252;
149 MY2   (* fail -> return 0, sonst die niederwertigsten 5 Bits v
        on IntuiMessage.code =
150 NV5      Nummer des Menüs *)
151 em0 PROCEDURE Itemid():CARDINAL; CODE -258;
152 1Z2   (* fail -> return 0, sonst die mittleren 6 Bits von Intu
        iMessage.code =
153 195      Nummer des Items *)
154 r40 PROCEDURE Subid():CARDINAL; CODE -264;
155 Qm2   (* fail -> return 0, sonst die höchstwertigen 5 Bits von
        IntuiMessage.code =
156 Jb5      Nummer des SubItems *)
157 p30 END M2Extintui.
158 I5  (* Hinweis: Die Schnittstellen-Module können jederzeit vom
    Anwender auf den
159 1d3    neuesten Stand gebracht und compiliert werden. Als Refer
       enz gilt 1.3Include.
160 zP     Sowohl in fcn.c als auch in M2 ist Gadget.gadgetID vom T
       yp SHORT(=INTEGER).
161 6Q     In 1.3 jedoch ist Gadget.gadgetID vom Typ USHORT(=CARDIN
       AL). *)
(C) 1989 M&T
```

Listing 6. (Schluß)

Meister der Tasten

Sind Sie in der Lage, Ihre Amiga-Tastatur blind zu bedienen? Tippen Sie schnell und fehlerlos? Wenn ja, können Sie die nächsten Zeilen getrost überlesen. Enthaltenn Ire Texyte jedoch jede menge Tipfeler, dann ist »Keyboard-Master« genau das Richtige für Sie.

Von Robert Kretzschmar

Besonders wenn Sie oft Briefe mit der Schreibmaschine oder einer Textverarbeitung verfassen oder auch hin und wieder Listings abtippen, ist es von großem Vorteil, blind schreiben zu können. Das erhöht nicht nur die Schreibgeschwindigkeit enorm, sondern hilft auch, Tippfehler zu vermeiden. Das Programm »Keyboard-Master« ermöglicht Ihnen das Erlernen des Zehn-Finger-Systems. Es stellt einen vollständigen, interaktiven Schreibmaschinen-Kurs dar, der in 20 Kapitel unterteilt ist. Die Spanne der Übungen reicht von einfachen »ASDF«-Aufgaben bis hin zu speziellen Fingerübungen für Programmierer.

Um den gewünschten Erfolg zu haben, ist es notwendig, möglichst regelmäßig zu trainieren und anfängliche Probleme durch Wiederholung der Übungen zu beseitigen. Wenn Sie das beachten, wird Ihnen dieses Lernprogramm sicherlich schnell die ersten Erfolgserlebnisse bescheren.

Die Lernerfolge mit »Keyboard-Master« sind wirklich erstaunlich. So mancher unserer Redakteure hat sich die Lektionen »reingezogen« und seine »Tipfelereien« enorm reduzieren können. Wer seine Kenntnisse im Maschineschreiben verbessern will, sollte sich das Programm unbedingt ansehen.

»Keyboard-Master« wurde in Amiga-Basic verfaßt. Trotzdem ist die Arbeitsgeschwindigkeit sehr hoch.

Das Listing

»Keyboard-Master« finden Sie auf der Programmservice-Diskette zu diesem Sonderheft. Ein Abdruck des Listings erschien uns unsinnig: Erstens ist das Programm sehr lang, und daher zum Abtippen kaum geeignet. Ein weiterer Grund sind die Übungsdateien. Wie gesagt, besteht der Kurs aus zwanzig Kapiteln mit verschiedenen Übungen. Diese müßten also, damit das Programm überhaupt lauffähig ist, ebenfalls eingegeben werden. Da dies sehr mühsam wäre, haben wir uns also entschlossen, »Keyboard-Master« im Heft nicht abzudrucken.

Sie finden also auf der Diskette zu diesem Sonderheft das Basic-Listing unter dem Namen »KeyboardMaster« zusammen mit den Files »Lektion 1.l« bis »Lektion 20.l« und der Übungsdatei »test.k«. Kopieren Sie alle Dateien auf eine Diskette, auf der sich Amiga-Basic befindet. Vor dem Start muß darauf geachtet werden, daß der deutsche Zeichensatz aktiviert ist (vom CLI aus mit »system/setmap d« aktivieren).

Die beiden Zeilen nach dem »CLEAR«-Befehl am Listing-Beginn enthalten zwei »Default«-Werte, die Sie Ihren Gegebenheiten anpassen können: »drive$« stellt die Laufwerksbezeichnung dar, die immer dann benutzt wird, wenn beim Laden der Kursdatei kein Laufwerk gesondert angegeben wird. Sinnvoll wäre hier, das Laufwerk (»df0:« oder »df1:«) anzugeben, von dem Sie gewöhnlich das Programm laden werden. Direkt darunter steht der voreingestellte Name der Kursdatei. In dieser Datei, die das Programm für jeden Benutzer gesondert anlegt, stehen die Informationen über die bearbeiteten Lektionen und die erzielten Ergebnisse. Der Name dieser Datei wird beim Programmstart immer erfragt. Wenn Sie nicht jedesmal denselben Filenamen eintippen wollen, drücken Sie einfach die RETURN-Taste und der von Ihnen im Programmtext voreingestellte Name wird verwendet. Als Filename wäre zum Beispiel Ihr Vorname sinnvoll. Das Programm hängt übrigens selbständig an den eingegebenen Namen ».k« an, um es als Kursdatei von anderen Files zu unterscheiden. Beim ersten Start des Programms wird dann die Kursdatei mit dem entsprechenden Namen auf Wunsch auf Diskette erzeugt. Auf der Diskette befindet sich übrigens bereits eine Kursdatei unter dem Namen »test.k« Wenn Sie das Listing entsprechend angepaßt haben, speichern Sie es erneut. Die Arbeit mit dem Programm kann nun beginnen.

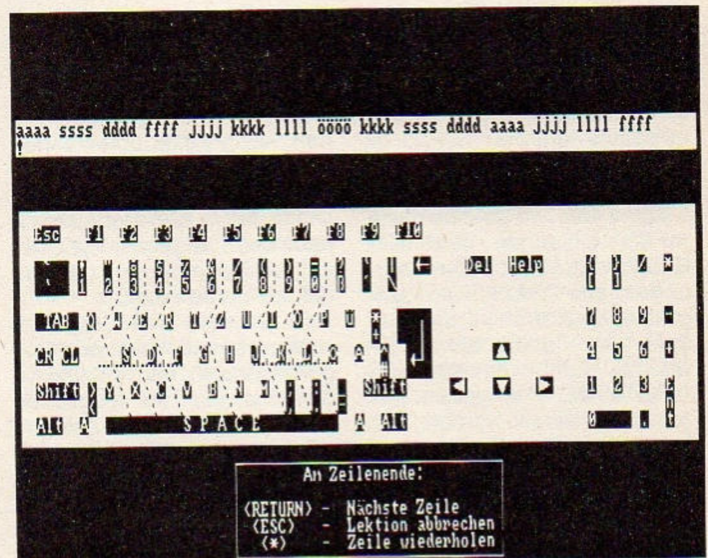


Bild 2. Die erste Übung hat begonnen

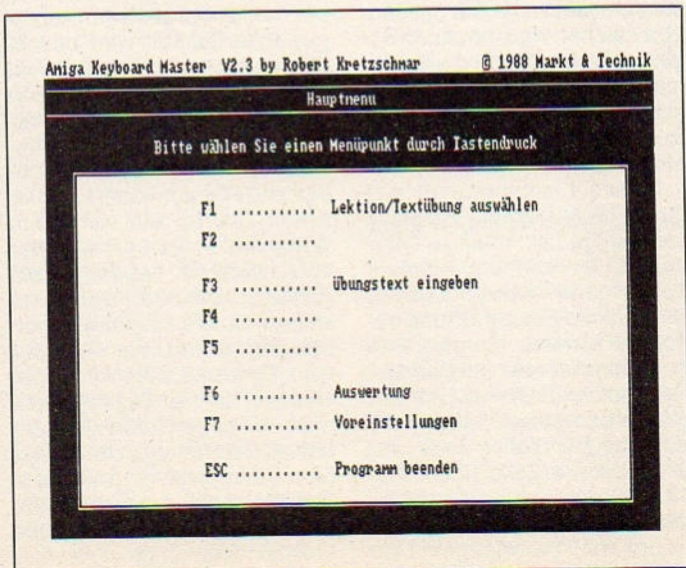


Bild 1. Das Hauptmenü von »Keyboard-Master«

Die ersten Schritte

Nach dem Start mit »RUN« werden Sie zuerst nach dem Namen der Kursdatei gefragt (siehe oben). Wollen Sie den vorgegebenen Namen verwenden, drücken Sie nur <RETURN>. Anschließend wird die Kursdatei geladen. Wenn sie unter diesem Namen noch nicht existiert, werden Sie gefragt, ob eine neue angelegt werden soll. Beantworten Sie mit »n«, können Sie den Namen nochmals eingeben. Tippen Sie »j« (was Sie beim ersten Start tun sollten), legt das Programm eine neue Datei an. Falls Sie eine bereits existie-

rende Kursdatei überschreiben wollen, müssen vor dem eigentlichen Filenamen drei Sterne und ein Leerzeichen eingegeben werden. Achtung: Alle Daten in dieser Datei gehen dabei verloren! Wichtig ist, daß Sie die Diskette mit der Kursdatei immer im Laufwerk lassen, solange Sie mit dem Programm arbeiten.

Sie befinden sich nun im Hauptmenü (Bild 1). Dabei wird Ihnen auffallen, daß das ganze Programm über die Tastatur gesteuert wird. Schließlich soll deren Bedienung ja erlernt werden. Zudem kann die Bedienung auch ohne Maus komfortabel und einfach sein.

Einige Menüpunkte, die im Moment noch nicht sinnvoll sind, fehlen und sind nicht wählbar. Im folgenden werden alle Menüpunkte der Reihe nach besprochen. Beginnen Sie jedoch nicht sofort, den Kurs durchzuarbeiten. Das sollten Sie erst dann tun, wenn Sie auch die Hinweise zur Durchführung des Kurses gelesen haben. Dies soll Sie aber nicht daran hindern, die Funktion des Programms beim Lesen der Bedienungsanleitung auszuprobieren.

<F1> - Lektion/Textübung wählen

Mit der Funktionstaste <F1> gelangen Sie vom Hauptmenü in ein Untermenü, in dem alle 20 Lektionen aufgelistet sind. Wählen Sie eine davon aus, indem Sie die jeweilige Zahl, gefolgt von <RETURN>, eingeben. Im Anschluß daran werden alle Lektionen in den Speicher geladen. Wenn Sie später weitere Lektionen anwählen, erfolgen keine weiteren Zugriffe auf die Diskette. Wenn Sie keine reguläre Lektion bearbeiten wollen, sondern einen beliebigen ASCII-Text üben wollen, drücken Sie statt einer Zahl nur die RETURN-Taste und Sie werden nach dem Filenamen des Textes gefragt. Tippen Sie den Namen des gewünschten Files ein. Nach dem Ladevorgang gelangen Sie wieder ins Haupt-Menü.

<F2> - Lektion starten

Je nachdem, ob Sie eine Lektion oder einen Übungstext geladen haben, hat dieser Menüpunkt den Titel »Lektion xx starten« oder »Textübung starten«. Nach Betätigen von <F2> werden zunächst einige Daten aus der Kursdatei geladen und der wichtigste Teil des Programms beginnt.

Auf dem Bildschirm wird ein Abbild einer Amiga-Tastatur, die an jene des Amiga 500 angelehnt ist, gezeichnet (Bild 2). Die Querlinien mitten im Tastenfeld sollen am Anfang helfen, alle Tasten mit dem richtigen Finger anzuschlagen. Die Hilfslinien trennen also den »Zuständigkeitsbereich« der einzelnen Finger voneinander. Die helle Linie in der Mitte stellt die Trennung zwischen rechter und linker Hand dar. Wenn in den Voreinstellungen (<F7>) das Zeichnen der Hilfslinien nicht angeschaltet ist, sind diese natürlich nicht zu sehen. Genauso verhält es sich mit der Hilfstastatur und der Anzeige der jeweils zu drückenden Taste (siehe unten).

Der Bildschirmaufbau benötigt einige Sekunden, jedoch nur beim ersten Aufruf dieses Programmteils. Danach ist die Grafik im RAM gespeichert. Ganz oben sehen Sie die Statuszeile, in der Sie lesen können, welche Lektion Sie gerade bearbeiten, welche Nummer die aktuelle Zeile hat und in welchem Teil der Lektion (Griff-, Wort- oder Satzübung) Sie sich im Moment befinden. Darunter wird die Anzahl der Fehler angezeigt, die Sie in der aktuellen Zeile gemacht haben. Noch weiter unten sehen Sie in einem farbigen Feld den Klartext der jeweiligen Zeile. Unter dieser befindet sich ein nach oben zeigender Pfeil, der auf das Zeichen zeigt, das eingetippt werden muß.

Im unteren Teil des Bildschirms befindet sich noch ein gelber Kasten, der als Hilfe für den Benutzer die möglichen Tasten am Zeilenende beinhaltet. Ist der Bilschirmaufbau abgeschlossen, ertönt ein Summen, das zeigt, daß es losgehen kann. Sie müssen nun Buchstaben für Buchstaben aus der angezeigten Textzeile abtippen. Der Pfeil wandert dabei von Zeichen zu Zeichen mit. Um Sie davon abzuhalten, spähende Blicke auf die Computertastatur zu werfen, werden die zu drückenden Tasten auf der Hilfstastatur hervorgehoben (sofern in der Voreinstellung) diese Funktion aktiviert wurde).

Machen Sie einen Fehler, hören Sie einen Piepston und der Pfeil verwandelt sich unter dem falsch getippten Zeichen in ein Ausrufungszeichen (»!«). Die Fehleranzeige wird um eins erhöht. Tippen Sie nun das Zeichen richtig, bekommt der Cursor (Pfeil) wieder sein normales Aussehen und Sie können weitertippen. Am Zeilenende angelangt erscheint an Stelle des Cursors ein Linkspfeil »—« und Sie können nun die RETURN-Taste drücken, um in die nächste Zeile zu gelangen, die ESC-Taste, um die Lektion ohne Speichern der Ergebnisse abzubrechen oder <*>, um die getippte Zeile nochmals einzugeben. In der Gesamtwertung werden dann nur jene Fehler, die beim weiteren Durcharbeiten der Zeile gemacht wurden, berücksichtigt.

Um die Funktionsweise auszutesten und sich daran zu gewöhnen, sollten Sie einmal eine Zeile abtippen, am Zeilenende aber nicht die RETURN-Taste drücken, sondern <ESC>. Schließlich handelte es sich nur um einen Versuch, und ein Speichern der Ergebnisse ist wohl kaum erwünscht.

Der Computer weiß alles...

Am Ende jeder Lektion werden die erzielten Ergebnisse, sowie allgemeine Angaben zur Lektion, am Bildschirm angezeigt (Bild 3) und in die Kursdatei gespeichert. So weiß das Programm immer, welche Lektion Sie schon durchgearbeitet haben, wieviel Fehler Sie dabei gemacht haben und wie schnell Sie beim Tippen waren. Genauso ist der Computer — und damit auch Sie — ständig im Bilde darüber, wieviel prozentual falsch gemacht wurde und wie viele Anschläge pro Minute Sie erreicht haben. Schließlich ist er auch noch darüber informiert, ob und wie oft Sie einzelne Lektionen wiederholt und sich dabei verbessert oder verschlechtert haben. Erwähnenswert ist hierbei auch die grafische Anzeige des Fehlerprozentsatzes. All das schauen Sie sich am besten erst an, wenn Sie mit dem eigentlichen Kurs ernsthaft beginnen wollen, denn die Bedienung dieses Programmteils erklärt sich von selbst.

<F3> - Übungstext eingeben

Dieses Lernprogramm gibt Ihnen nicht nur die Möglichkeit, fertige Lektionen zu bearbeiten, es können vielmehr auch eigene Texte eingegeben werden, die Sie zur Übung abtippen können. Dieser Programmteil ist ebenfalls weitgehend selbsterklärend. Es bleibt nur zu bemerken, daß Sie die Eingabe der Zeilen (maximal 20 Zeilen, je Zeile höchstens 78 Zeichen) mit einer Leerzeile abschließen müssen, um ins Hauptmenü zu gelangen. Dort werden Sie bemerken, daß unter »F2« nun »Textübung starten« angeführt ist. Schließlich wollen Sie jetzt Ihren Text bearbeiten, nicht die Lektion. Dasselbe gilt beim Laden eines Übungstextes (siehe oben).

<F4> - Lektions-/ Übungstext anzeigen

Über diesen Menüpunkt wird der aktuelle Lektionstext beziehungsweise der eingegebene oder geladene Übungstext angezeigt. Dabei fallen hier die violetten Zahlen am Zeilenanfang auf. Diese zeigen an, wie oft jede Zeile abgetippt werden muß.

Bei der Eingabe eines Übungstextes wird übrigens automatisch eine »1« vor jede Zeile geschrieben. Auch wenn Sie eine ASCII-Datei laden (Menüpunkt 1), wird jede Zeile, die nicht mit einer Ziffer beginnt, mit einer »1« am Zeilenanfang ergänzt. Bei den Lektionstexten können Sie mit einem ASCII-Editor die Anzahl der Wiederholungen leicht ändern. Beachten Sie jedoch, daß nur die erste Ziffer als Wiederholungszahl erkannt wird. Diesen Programmteil verlassen Sie, indem Sie eine beliebige Taste betätigen.

<F5> - Übungstext speichern

Dieser Menüpunkt kann nur dann gewählt werden, wenn Sie zuvor einen Übungstext geladen oder eingegeben haben. Das Programm fragt Sie nach dem gewünschten Filenamen. Wollen Sie den Text doch nicht speichern, drücken Sie nur <RETURN>.

<F6> - Auswertung

In diesem Programmteil werden die Daten, die die Kursdatei über die einzelnen Lektionen enthält, angezeigt (Bild 4). Lektionen, die noch nicht bearbeitet worden sind, werden dabei dunkel dargestellt. Die angewählte Lektion wird besonders gekennzeichnet. Das Feld »Differenz FQ« enthält, wenn die Lektion mehr als einmal bearbeitet worden ist, die Differenz des neueren Fehlerquotienten (FQ) zum älteren. Hier sehen Sie, ob Sie sich beim Wiederholen verbessert oder verschlechtert haben. Direkt daneben befindet sich eine weitere Spalte, die Ihnen noch unbekannt ist: Unter »Punkte« wird für jede Lektion eine bestimmte Zahl angegeben, die etwa mit einer Note vergleichbar ist. Bei Punktewerten unter 50 werden dahinter drei Ausrufungszeichen geschrieben, was bedeutet, daß Sie diese Lektion unbedingt wiederholen sollten. Werte über 100 sind sehr gut. In die Punktwertung

Amiga Keyboard Master V2.3 by Robert Kretzschmar © 1988 Markt & Technik

ENDE von Lektion 1

Ergebnis

Anschläge:	662
Fehler:	49
Fehlerquote:	7.40 %
benötigte Zeit:	9 Minuten, 35 Sekunden
Anschläge pro Minute:	69
Wiederholung der Lektion:	keine

Drücken Sie eine Taste

Bild 3. Die Auswertung gibt Aufschluß über den Lernerfolg und Ihren momentanen Leistungsstand

Amiga Keyboard Master V2.3 by Robert Kretzschmar © 1988 Markt & Technik

Auswertung (Kursdatei df0:test)

Lektion	Wiederh.	Fehlerquotient	Anschl./min	Differenz FQ	Punkte
1	0 mal	7.40 %	69		90
2	0 mal	5.47 %	177		112
3					
4					
5					
6	0 mal	3.81 %	150		111
7					
8					
9					
10					
11					
12					
13					
14	0 mal	6.58 %	118		97
15					
16					
17					
18					
19					
20					

Taste drücken

Bild 4. Durch das Anlegen einer Kursdatei auf Diskette wissen Sie ständig über Ihren »Wissens-Status« bescheid

geht neben einigen anderen Faktoren hauptsächlich der Fehlerquotient ein.

<F7> - Voreinstellungen

Mit <F7> gelangen Sie in den Programmteil, in dem Sie verschiedene Voreinstellungen (vergleichbar mit Preferences der Workbench) ändern können. Diese Änderungen werden in die Kursdatei geschrieben und bei jedem erneuten Laden der Kursdatei im Programm berücksichtigt.

Sie haben sechs Möglichkeiten, Einstellungen und damit den Programmablauf zu verändern. Es handelt sich dabei um »Schalter«, wobei durch Druck auf die links neben der jeweiligen Bezeichnung stehende Zahlentaste zwischen zwei Stellungen hin- und hergeschaltet werden kann. Bei einigen Schalterkombinationen nimmt das Programm selbständig Änderungen der Stellungen vor, das heißt es verhindert unlogische Kombinationen. Kommen wir nun zur Bedeutung der einzelnen Schalter:

Wenn Schalter 1 ausgeschaltet ist, wird die Hilfstastatur nicht mehr angezeigt.

Schalter 2 ist für die Hervorhebung der jeweils zu drückenden Tasten auf der gezeichneten Hilfstastatur zuständig.

Mit Schalter 3 wird bestimmt, ob die Textzeile immer angezeigt wird oder nicht. Es ist sinnvoll, diese auszuschalten, wenn man sich voll auf die Griffe und die Hilfstastatur konzentrieren möchte.

Ist Schalter 4 ausgeschaltet, wird das Zeichnen der Hilfslinien auf der Tastatur unterdrückt, da diese mit der Zeit störend wirken können.

Wenn Sie das automatische Wiederholen einzelner Zeilen unterdrücken wollen, müssen Sie Schalter 5 ausschalten. Dies ist dann sinnvoll, wenn Sie eine Lektion schon oft bearbeitet haben und diese nur kurz auffrischen wollen.

Mit dem sechsten und letzten Schalter wählen Sie schließlich das Default-Laufwerk (df0: oder df1:), das immer dann verwendet wird, wenn bei Filenamen keine Laufwerksangabe gesondert angegeben wird. Auch die Lektionstexte werden stets von diesem Laufwerk geladen. Bitte verwechseln Sie diese Einstellung nicht mit dem Default-Laufwerk, das Sie vor dem ersten Programmstart im Listing direkt eingestellt haben. Dieses Laufwerk gilt nur für die Kursdatei, ansonsten hat das in diesem Programmteil einzustellende Laufwerk Priorität.

Mit <ESC> werden eventuelle Änderungen in die Kursdatei gespeichert und Sie kommen zurück ins Hauptmenü.

ESC - Programm beenden

Mit <ESC> im Haupt-Menü verlassen Sie das Programm.

Der Kurs beginnt

Ich hoffe, Sie haben nun die ersten Gehversuche mit Keyboard-Master hinter sich und brennen darauf, endlich den Kurs zu beginnen.

Dazu greifen Sie zunächst, ohne das Programm zu starten, die Grundstellung, die Sie in Tabelle 1 (Seite 35) finden. Diese Fingerstellung ist sehr wichtig, prägen Sie sich diese ein und versuchen Sie mehrmals, diese zu finden, ohne auf die Tastatur zu sehen.

Sie sollten nun das Programm erneut starten und einen beliebigen Filenamen für die Kursdatei (etwa Ihren Vornamen) eingeben. Wenn Sie bei den ersten Versuchen schon eine Datei unter diesem Namen erstellt haben, geben Sie drei Sterne und ein Leerzeichen vor dem eigentlichen Namen ein, so daß in jedem Fall eine völlig neue Kursdatei angelegt wird. Nun sehen Sie sich die aktuellen Voreinstellungen des Programms mit <F7> an. Ändern sollten Sie zunächst allerdings nur das Default-Laufwerk, um es Ihren Gegebenheiten anzupassen. Gehen Sie zurück ins Hauptmenü und drücken Sie <F1>. Dort geben Sie »1« gefolgt von <RETURN> ein. Nach dem Ladevorgang können Sie die Lektion starten. Bitte sehen Sie sich aber vorher die folgenden zehn Grundregeln (das sollten Sie zur besseren Einprägung möglichst vor jedem Start einer Lektion tun) an:

1. Arbeiten Sie möglichst regelmäßig mit dem Programm.
2. Bearbeiten Sie an einem Tag nur eine Lektion.
3. Wiederholen Sie jede Lektion am besten mindestens einmal.
4. Lassen Sie sich nicht von anfänglichen Schwierigkeiten entmutigen.
5. Greifen Sie die Grundstellung jedesmal blind.
6. Sehen Sie **nie** auf die Tastatur, orientieren Sie sich an der Hilfstastatur am Bildschirm.
7. Der Anschlag erfolgt kurz und ruckartig aus dem Muskel heraus.
8. Gehen Sie nach jedem Anschlag schnell in die Grundstellung zurück.
9. Tippen Sie langsam, aber gleichmäßig im Takt.
10. Halten Sie die Handgelenke möglichst hoch. Die Hand darf nicht flach auf der Tastatur liegen, sie muß angewinkelt sein.
11. Es kann nicht schaden, wenn Sie Ihre Finger mit einer speziellen Gymnastik trainieren.

Noch ein Hinweis zu Lektion 12: Beachten Sie, daß eine zu »shiftende« Taste, die sich im rechten Teil der Tastatur befindet, immer zusammen mit der linken SHIFT-Taste (und umgekehrt) angeschlagen wird. Das Programm unterstützt Sie dabei, indem immer die richtige zu drückende SHIFT-Taste hervorgehoben wird.

Haben Sie sich all dies zu Herzen genommen, kann es losgehen: Drücken Sie <F2> und tippen Sie nun alle Zeilen der Lektion ein. Schließen Sie jede Zeile mit <RETURN> ab. Werden Sie nicht bei jedem Fehler nervös. Errare humanum est! (lat.: Irren ist menschlich!) Bleiben Sie konzentriert und denken Sie daran, daß die ersten Lektionen die schwierigsten sind.

Sind Sie mit der Lektion fertig, werden Ihnen die Ergebnisse gezeigt. Hinter Fehler, Fehlerquotient und Anschläge pro Minute stehen je zwei Zahlen, jedoch nur dann, wenn Sie die Lektion insgesamt mindestens zweimal bearbeitet haben. Die zweite Zahl stellt jeweils die Differenz zum Ergebnis des letzten Mals dar. Die Anzeige der Anschläge je Minute sollten Sie bei den ersten Lektionen einfach übersehen. Sie sind beim Lernen völlig unwichtig und werden in der

Punktewertung kaum berücksichtigt. Wichtig ist, daß Sie möglichst gleichmäßig tippen und wenige Fehler machen. Wenn Sie nun vom Hauptmenü mit <F6> die Auswertung aufrufen, können Sie sich die erreichte Punktzahl ansehen. Sie sollten diese jedoch nicht mit Schulnoten vergleichen. Denken Sie daran, daß Sie hier völlig ohne Druck lernen. Die Punkte und auch die Tippgeschwindigkeit sollen Ihnen vor allem als Vergleich dienen, wenn Sie schon einige Lektionen hinter sich haben. Punktwerte unter »50« sind jedoch ein Warnzeichen dafür, daß Sie diese Lektion wiederholen müssen. Wenn Sie mehr als »80« Punkte erreichen, können Sie sehr zufrieden sein, Werte über »100« oder gar über »120« sind sehr gut.

M. Jobst/kn

Linke Hand:	a	kleiner Finger
	s	Ringfinger
	d	Mittelfinger
	f	Zeigefinger
Rechte Hand:	j	Zeigefinger
	k	Mittelfinger
	l	Ringfinger
	ö	kleiner Finger

Tabelle 1. Die Grundstellung beim Maschineschreiben ist die Basis für alle Übungen

Sounds im Eigenbau

Mit dem »Sound-Effekte-Editor« erzeugen Sie komfortabel alle Geräusche. Vom einfachen akustischen Hinweis auf Eingabefehler bis zum Kanonendonner in Spielen reicht das Spektrum. Der Clou: Die Sounds werden als Quellcode gespeichert und sind so leicht für eigene Programme zu verwenden.

Von Arno Gölzer

Wenn auf dem Mond riesige Meteore einschlagen und dabei gewaltige Krater hinterlassen, so geschieht das vollkommen lautlos. Schuld daran ist die fehlende Lufthülle. Der Schall ist eine hörbare oder zumindest nachweisbare Schwingung, die sich nur in einem Medium, beispielsweise in der Luft oder im Wasser, verbreitet. Jedes Geräusch hat seine eigene, individuelle Schwingung und kann daher eindeutig identifiziert werden. Aber auch eine — zumindest annähernde — mathematische Beschreibung ist möglich. Wir unterscheiden zwischen vier Grundschwingungen:

- Sinus
- Rechteck
- Dreieck
- Sägezahn

Die Klangfarbe eines Tones machen aber im wesentlichen die sogenannten Oberschwingungen aus. Es handelt sich um mehr oder weniger große Abweichungen von der eigentlichen Wellenform. Der Amiga läßt uns in bezug auf das Aussehen der Wellenform völlig freie Hand. Somit ist es möglich, beliebige Geräusche künstlich zu erzeugen - wir müssen nur die Wellenform entsprechend beschreiben. Diese Aufgabe wird mit dem Sound-Effekte-Editor (SEEd) (Listing 1) zum Kinderspiel.

Nach dem Eingeben des Programms mit dem Checksummer (Seite 159) und dem Compilieren starten Sie SEEd vom CLI aus, indem Sie den Programmnamen eingeben und <RETURN> drücken. Haben Sie ein Programm-Icon für SEEd entworfen, können Sie den Soundgenerator auch von der Workbench aus starten. Wichtig ist in jedem Fall, daß sich im Verzeichnis »Libs« die Mathematik-Libraries befinden (mathtrans.library).

Von der Welle zum Klang

Nach dem Programmstart öffnet sich ein Fenster, das in zwei Felder unterteilt ist (Bild 1):

1. Das Bedienungsfeld belegt das rechte Viertel des Bildschirms. Es besteht aus insgesamt zwölf Reglern (Gadgets). Lesen Sie im folgenden deren Funktion:

Großer Schieberegler links:
Grobeinstellen der Frequenz. Schieben Sie den Regelknopf nach unten, wenn Sie niedrigere Frequenzen einstellen möchten.

Großer Schieberegler rechts:
Feineinstellen der Frequenz. Mit den beiden Frequenzreglern können Sie über 65000 Frequenzen einstellen!

Kleiner Schieberegler links:
Hier stellen Sie ein, wie oft der eingestellte Klang wiederholt werden soll.

Kleiner Schieberegler rechts:
Die Lautstärke.

Null:
Wellenform auf Null stellen (waagerechter Strich erscheint).

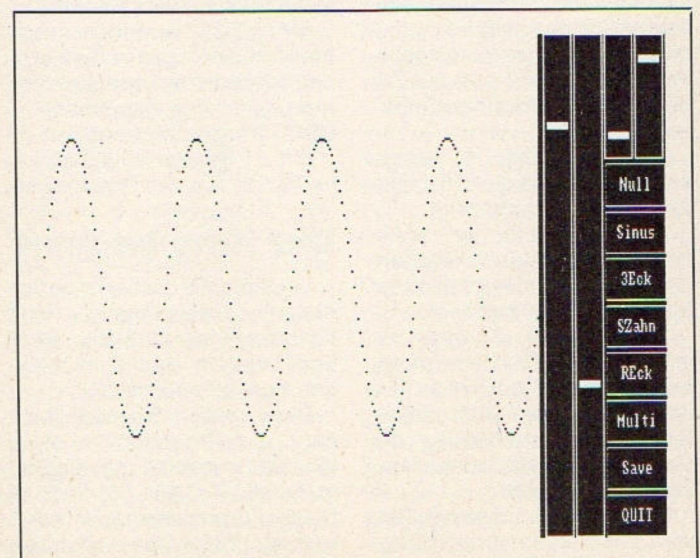


Bild 1. Das SEEd-Fenster. Im linken Teil bearbeiten Sie den gewünschten Sound, rechts befinden sich alle Regler.

Sinus:
Einstellen der Wellenform auf Sinus.
3Eck:
Eine Dreieckschwingung wird eingestellt.
SZahn:
Vorwahl der Sägezahnschwingung.
REck:
Die Rechteckschwingung einstellen.

Die voreingestellten Schwingungen können jetzt beliebig verändert werden (siehe unten). Allerdings ist in diesem Zustand der Klang nicht immer optimal. Verbessern Sie die Qualität, indem Sie mehrere Schwingungen im Zeichenfeld unterbringen:
Multi:
Eingestellte Schwingung verdoppeln.
Save:
»Geräusch« auf Diskette sichern. Ein Requester öffnet sich. Geben Sie den Filenamen an und klicken Sie das Gadget mit der Aufschrift »Save« an (oder »Break«, wenn Sie lieber nicht speichern möchten).
Quit:
Mit diesem Schalter brechen Sie das Programm ab.

Im weiß hinterlegten Zeichenfeld können Sie vorgefertigte Schwingungen verändern

Sounds zeichnen mit der Maus

oder aber völlig neue entwerfen. Drücken Sie die linke Maustaste und »zeichnen« Sie Ihren Soundeffekt. Die besten Ergebnisse erhalten Sie, wenn Sie die Höhe des Zeichenfeldes möglichst voll ausnutzen. Im anderen Fall sind die Schwingungen nur schlecht hörbar.

Weiterhin ist es von Vorteil, wenn Schwingungsanfang und -ende auf gleicher Höhe liegen. Nutzen Sie auch hier die Multi-Funktion zur Verbesserung der Soundqualität. Sie hören den Klang, wenn Sie die rechte Maustaste betätigen. Bei den Grundschwingungen ist allerdings selten ein Ton zu hören. Keine Angst, das Programm funktioniert trotzdem fehlerfrei. Verdoppeln Sie die Grundschwingung mehrfach, entlocken Sie dem Amiga bereits einen Klang.

Begrenzen der Wellenform

Mit den orangefarbenen Markierungen im Zeichenfeld können Sie den Anfang und das Ende der Wellenform selbst festlegen. Bedienen Sie die Markierungen mit dem Ziffern-Block der Tastatur:
<7> Startmarkierung nach links,
<9> Startmarkierung nach rechts,
<8> Startmarkierung auf Start.

Da das Datenfeld auf einem Langwort (Longword) beginnen muß, erfolgt die Bewegung der Startmarkierung in 4er-Sprüngen.
<4> Endemarkierung nach links,

Einbindung in C-Programme

<6> Endemarkierung nach rechts,
<5> Endemarkierung auf Ende.

Das Ende des Datenfeldes ist nicht an bestimmte Speicherbereiche gebunden. Daher ist hier noch eine Feineinstellung in Einerschritten möglich:
<1> Endemarkierung einen Schritt nach links,
<3> Endemarkierung einen Schritt nach rechts,
<2> Endemarkierung auf Ende.

Das Einbinden eines SEEd-Sounds in C-Programme ist sehr einfach, da SEEd den Sound als C-Quellcode ablegt. Folgende Schritte sind nötig:
- Binden Sie die SEEd-Datei als Headerfile in Ihr eigenes Programm ein.
- Rufen Sie zu Beginn Ihres eigenen Programms die Funktion SoundOn() auf. Die Funktion liefert im Fehlerfall den Wert 1.

Ihr Sound ist nun initialisiert und Sie können ihn jederzeit und beliebig oft abspielen:
- »Play()« spielt den Soundeffekt ab.
- Rufen Sie vor Beendigung Ihres Programms die Funktion »SoundOff()« auf. Sie gibt alle für den Soundeffekt notwendigen Ressourcen wieder frei.

Play() und SoundOff() liefern keinen Wert.

Hier ein kleines Beispiel:

```
#include "SoundName"

main()
{
        if(!SoundOn()){
            Play();
            SoundOff();
        }
}
```

Aztek-C-Besitzer übersetzen dieses Beispiel so:

```
cc NAME.c +L
ln NAME.o -lc32
```

Anstelle von »NAME« muß natürlich der Name Ihres Programms stehen. Programmierer anderer Programmiersprachen profitieren immer noch von den gesicherten Sounddaten. Deshalb enthält Listing 2 für den Interessierten zum Schluß noch das dokumentierte Listing einer SEEd-Sounddatei. Mit dem SEEd geben wir Ihnen ein wertvolles Werkzeug zur Erzeugung von Sounds in eigenen Programmen in die Hand. Ein Werkzeug, das die Soundprogrammierung ungemein erleichtert und das Sie bestimmt bald nicht mehr missen wollen.

Arno Gölzer/kn

Programmname:	SEEd
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztek V3.4
Aufrufe:	cc SEED.c +L ln SEEd.o -lm32 -lc32
Bemerkung:	benötigt Math.-Libraries

Programmautor: Arno Gölzer

```
/*
**     Gölzy's SEEd.c
**        08.88
*/
#include <math.h>
#include <stdio.h>
#include <exec/types.h>
#include <intuition/intuition.h>
#include <graphics/gfx.h>
#include <devices/audio.h>
#include <exec/memory.h>
#define GH (0xFFFF/255)
#define GL (0xFFFF/255)
#define GC (0xFFFF/64)
#define GV (0xFFFF/64)
#define ENDE (((Ende-Start)/2)+1)
#define MOUSX (win->MouseX)
#define MOUSY (win->MouseY)
#define GAD_ID (((struct Gadget *)ADDR)->GadgetID)
APTR ADDR;
ULONG CLASS, CODE;
UBYTE *Data,buffer=15,name[31],undo[31],Titel[]=(UBYTE *)"-
*- Gölzy's SEEd -*-";
UWORD l,h,p,c,v;
struct Image HDummy,LDummy,CDummy,VDummy;
SHORT Start=0,Ende=510;
SHORT OKpairs[]       ={ 0,0,  61,0,  61,21, 0,21, 0,0  };
SHORT Reqpairs[]      ={ 0,0, 246,0, 246,67, 0,67, 0,0  };
SHORT Spairs[]        ={ 0,0, 210,0, 210,11, 0,11, 0,0  };
struct Border OKRand  ={ -1, -1, 1,0, JAM1, 5,&OKpairs[0],
 NULL      };
struct Border ReqR    ={ 2, 1, 1,0, JAM1, 5,&Reqpairs[0],
NULL      };
struct Border SRand   ={ -1, -2, 1,0, JAM1, 5,&Spairs[0],
NULL      };
struct IntuiText OKT2  ={ 1,0,JAM1,14,6,NULL,(UBYTE *)"QUIT
",NULL      };
struct IntuiText OKText={ 3,0,JAM1,15,7,NULL,(UBYTE *)"QUIT
",&OKT2     };
struct IntuiText SaText={ 1,0,JAM1,14,6,NULL,(UBYTE *)"Save
",NULL      };
struct IntuiText MText ={ 1,0,JAM1,10,6,NULL,(UBYTE *)"Mult
i",NULL     };
struct IntuiText RText ={ 1,0,JAM1,14,6,NULL,(UBYTE *)"REck
",NULL      };
struct IntuiText ZText ={ 1,0,JAM1,10,6,NULL,(UBYTE *)"SZah
n",NULL     };
struct IntuiText DText ={ 1,0,JAM1,14,6,NULL,(UBYTE *)"3Eck
",NULL      };
struct IntuiText SText ={ 1,0,JAM1,10,6,NULL,(UBYTE *)"Sinu
s",NULL     };
struct IntuiText NText ={ 1,0,JAM1,14,6,NULL,(UBYTE *)"Null
```

Listing 1. Mit dem Sound-Effekte-Editor (SEEd) entwerfen Sie Geräusche mit der Maus

```
       ",NULL      };
41 aL  struct IntuiText ReqT2 ={ 1,0,JAM1,45,6,NULL,Titel,NULL
              };
42 yx  struct IntuiText ReqT  ={ 3,0,JAM1,46,7,NULL,Titel,&ReqT2
              };
43 i3  struct IntuiText RCaT  ={ 1,0,JAM1,10,6,NULL,(UBYTE *)"Brea
       k",NULL      };
44 FC  struct IntuiText RTGT  ={1,0,JAM1,70,-10,NULL,(UBYTE *)"Fil
       ename:",NULL};
45 qZ  struct PropInfo PHInfo ={ FREEVERT| AUTOKNOB,0,0,      0,GH,0
       ,0,0,0,0,0  };
46 V7  struct PropInfo PLInfo ={ FREEVERT| AUTOKNOB,0,GL*255,0,GL,0
       ,0,0,0,0,0  };
47 Vw  struct PropInfo CInfo  ={ FREEVERT| AUTOKNOB,0,GC*64, 0,GC,0
       ,0,0,0,0,0  };
48 be  struct PropInfo VInfo  ={ FREEVERT| AUTOKNOB,0,0,      0,GV,0
       ,0,0,0,0,0  };
49 9C  struct StringInfo Sinfo={ name,undo,0,31,0,0,0,0,0,0,0,0,NU
       LL         };
50 uM  struct Gadget RTGad    ={ NULL,20,30,210,10,GADGHCOMP,
51 cU4      GADGIMMEDIATE| RELVERIFY,STRGADGET| REQGADGET,
52 wb       (APTR)&SRand,NULL,&RTGT,0,(APTR)&Sinfo,14,NULL
          };
53 6d0 struct Gadget RCa      ={ &RTGad,20,45,60,20,GADGHCOMP,
54 oc4      ENDGADGET| GADGIMMEDIATE| RELVERIFY,BOOLGADGET| REQGADGET,
55 e4       (APTR)&OKRand,NULL,&RCaT,0,0,13,NULL
          };
56 we0 struct Gadget ROk      ={ &RCa,170,45,60,20,GADGHCOMP,
57 rf4      ENDGADGET| GADGIMMEDIATE| RELVERIFY,BOOLGADGET| REQGADGET,
58 W3       (APTR)&OKRand,NULL,&SaText,0,0,12,NULL
          };
59 mu0 struct Gadget OKGad    ={ NULL,579,235,60,20,GADGHCOMP,
60 r84      GADGIMMEDIATE| RELVERIFY,BOOLGADGET,
61 JP       (APTR)&OKRand,NULL,&OKText,0,0,11,NULL
          };
62 6d0 struct Gadget SaGad    ={ &OKGad,579,211,60,20,GADGHCOMP,
63 uB4      GADGIMMEDIATE| RELVERIFY,BOOLGADGET,
64 Uz       (APTR)&OKRand,NULL,&SaText,0,0,10,NULL
          };
65 M00 struct Gadget MGad     ={ &SaGad,579,187,60,20,GADGHCOMP,
66 xE4      GADGIMMEDIATE| RELVERIFY,BOOLGADGET,
67 TV       (APTR)&OKRand,NULL,&MText,0,0,9,NULL
          };
68 Ma0 struct Gadget RGad     ={ &MGad,579,163,60,20,GADGHCOMP,
69 OH4      GADGIMMEDIATE| RELVERIFY,BOOLGADGET,
70 ek       (APTR)&OKRand,NULL,&RText,0,0,8,NULL
          };
71 fV0 struct Gadget ZGad     ={ &RGad,579,139,60,20,GADGHCOMP,
72 3K4      GADGIMMEDIATE| RELVERIFY,BOOLGADGET,
73 v8       (APTR)&OKRand,NULL,&ZText,0,0,7,NULL
          };
74 cu0 struct Gadget DGad     ={ &ZGad,579,115,60,20,GADGHCOMP,
75 6N4      GADGIMMEDIATE| RELVERIFY,BOOLGADGET,
76 E4       (APTR)&OKRand,NULL,&DText,0,0,6,NULL
          };
77 Rs0 struct Gadget SGad     ={ &DGad,579,91,60,20,GADGHCOMP,
78 9Q4      GADGIMMEDIATE| RELVERIFY,BOOLGADGET,
79 jn       (APTR)&OKRand,NULL,&SText,0,0,5,NULL
          };
80 vy0 struct Gadget NGad     ={ &SGad,579,67,60,20,GADGHCOMP,
81 CT4      GADGIMMEDIATE| RELVERIFY,BOOLGADGET,
82 aY       (APTR)&OKRand,NULL,&NText,0,0,4,NULL
          };
83 Ys0 struct Gadget VGad     ={ &NGad,608,0,30,64,0,0,PROPGADGET,
84 el4      (APTR)&VDummy,NULL,NULL,0,(APTR)&VInfo,3,NULL
          };
85 SD0 struct Gadget CGad     ={ &VGad,577,0,30,64,0,0,PROPGADGET,
86 p04      (APTR)&CDummy,NULL,NULL,0,(APTR)&CInfo,2,NULL
          };
87 H30 struct Gadget PGadL     ={ &CGad,546,0,30,256,0,0,PROPGADGET
       ,
88 QH4      (APTR)&LDummy,NULL,NULL,0,(APTR)&PLInfo,1,NULL
          };
89 u10 struct Gadget PGadH     ={ &PGadL,515,0,30,256,0,0,PROPGADGE
       T,
90 mo4      (APTR)&HDummy,NULL,NULL,0,(APTR)&PHInfo,0,NULL
          };
91 ji0 struct Requester Req={ NULL,10,10,250,70,0,0,
92 KC4      &ROk,&ReqR,&ReqT,0,0,NULL,NULL,NULL,NULL,NULL
          };
93 a00 struct NewWindow BW     ={
94 Gw4      0,0,640,256,-1,-1,VANILLAKEY| MOUSEBUTTONS| GADGETUP,
95 K2       ACTIVATE| BORDERLESS| RMBTRAP,&PGadH,NULL,NULL,NULL,NULL,
96 JJ       0,0,0,0,WBENCHSCREEN
          };
97 w40 struct Library          *OpenLibrary();
98 Hq  struct IntuiMessage     *GetMsg();
99 b1  struct MsgPort          *SPort,*CreatePort();
100 oq struct Device           *SDev,*OpenDevice();
101 vE struct IntuitionBase    *IntuitionBase;
102 3D struct GfxBase          *GfxBase;
103 Ns struct MathTransBase    *MathTransBase;
104 Np struct Window           *win,*OpenWindow();
105 Ls struct RastPort         *RP;
106 y0 struct IOAudio          Sound;
107 Bd VOID closew()                /* zum Programmende alle Re
   sourcen freigeben */
108 g9 {
109 5R4     if(!SDev)         CloseDevice(&Sound);
110 g0      if(SPort)         DeletePort(SPort,sizeof(struct MsgPor
   t));
111 iu      if(Data)          FreeMem(Data,256*sizeof(UBYTE));
112 C3      if(win)           CloseWindow(win);
113 sS      if(MathTransBase) CloseLibrary(MathTransBase);
114 de      if(GfxBase)       CloseLibrary(GfxBase);
115 LA      if(IntuitionBase) CloseLibrary(IntuitionBase);
116 Jh      exit(0);
117 t00 }
118 6H VOID openw()             /* Programmstart: Libraries, Windo
   w, ChipMemory ... */
119 rK {
120 zi4     UBYTE *AllocMem();
121 GP      if(!(IntuitionBase=(struct IntuitionBase *)
122 BuC             OpenLibrary("intuition.library",0)))      close
                w();
123 KB4     if(!(GfxBase=(struct GfxBase *)
124 sVC             OpenLibrary("graphics.library",0)))       close
                w();
125 h24     if(!(MathTransBase=(struct MathTransBase *)
126 foC             OpenLibrary("mathtrans.library",0)))      close
                w();
127 6U4     if(!(win=OpenWindow(&BW)))                         close
        w();
128 hE      if(!(Data=AllocMem(256*sizeof(UBYTE),MEMF_CHIP))) close
        w();
129 zv      if(!(SPort=CreatePort("SPort",0)))                 close
        w();
130 Tz      Sound.ioa_Request.io_Message.mn_ReplyPort  =SPort;
131 uS      Sound.ioa_Request.io_Message.mn_Node.ln_Pri=10;
132 08      Sound.ioa_Data   =&buffer;
133 Rd      Sound.ioa_Length=(ULONG)sizeof(buffer);
134 FO      if(SDev=OpenDevice(AUDIONAME,0,&Sound,0))          close
        w();
135 i7      Sound.ioa_Request.io_Command=CMD_WRITE;
136 QU      Sound.ioa_Request.io_Flags  =ADIOF_PERVOL| IOF_QUICK;
137 BV      RP =win->RPort;
138 EJ0 }
139 jz ULONG GetMessage()                                    /* Mes
   sagePort auslesen */
140 Cf {
141 yg4     struct IntuiMessage *mes;
142 Ko      CLASS=CODE=0;
143 KE      ADDR=NULL;
144 yD      if(mes=GetMsg(win->UserPort)){        CLASS=mes->Class
        ;
145 VPf                                           CODE=mes->Code;
146 xs                                            ADDR=mes->IAddres
                                              s;
147 NT                                            ReplyMsg(mes);
148 0t4     }
149 N0      return(CLASS);
150 Qv0 }
151 Eh VOID SetPoint(x,y)          /* Punkt (incl. Linie) im Zei
   chenfeld ausgeben */
152 8f4     SHORT x;
153 Ky      UBYTE y;
154 Qt0 {
155 Gm4     SHORT col;
156 hE      col=2*((x==Ende)||(x==Start));
157 6E      Data[x/2]=127-y;
158 90      SetDrMd(RP,JAM1);
159 4R      SetAPen(RP,1+col);
160 Pe      Move(RP,x,0);
161 yb      Draw(RP,x,255);
```

```
    Move(RP,x+1,0);
    Draw(RP,x+1,255);
    SetAPen(RP, 2);
    WritePixel(RP,x,y);
    WritePixel(RP,x+1,y);
}
VOID NewW()           /* neue Werte für Period, Cycle
 und Volume lesen */
{
    l=PLInfo.VertPot/GL;
    h=PHInfo.VertPot/GH;
    p=256*h+l;
    p=(p<124?124:p);
    c=64-CInfo.VertPot/GC;
    c=(c==0?1:c);
    v=64-VInfo.VertPot/GV;
}
Play()
* Klang abspielen */
{
    NewW();
    Sound.ioa_Data  =Data+Start/2;
    Sound.ioa_Length=ENDE;
    Sound.ioa_Period=p;
    Sound.ioa_Cycles=c;
    Sound.ioa_Volume=v;
    BeginIO(&Sound);
    WaitIO(&Sound);
}
VOID Save()
{
  COUNT i,j;
  FILE *file;
  NewW();
  if(file=fopen(name,"w")){
     fprintf(file,"/*   -*- %s - erstellt mit Gölzy's SEEd
     -*-   */\n\n",name);
     fprintf(file,"#include <devices/audio.h>\n");
     fprintf(file,"#include <exec/memory.h>\n");
     fprintf(file,"#define Play(); BeginIO(&Sound); WaitIO
     (&Sound);\n\n");
     fprintf(file,"UBYTE buffer=15,*SData,SD[%d]={\n\t",END
     E);
     for(i=Start/2,j=0;i<=Ende/2;i++,j++){
         fprintf(file,"%3d",Data[i]);
         if(i!=Ende/2)    fprintf(file,",");
         if(j==15){       fprintf(file,"\n\t");
                          j=-1;
         }
     }
     fprintf(file,"\n};\n");
     fprintf(file,"extern struct MsgPort\t*CreatePort();\n"
     );
     fprintf(file,"extern struct Device\t*OpenDevice();\n")
     ;
     fprintf(file,"struct MsgPort\t\t*SPort;\n");
     fprintf(file,"struct Device\t\t*SDev;\n");
     fprintf(file,"struct IOAudio\t\t Sound;\n\n");
     fprintf(file,"VOID SoundOff()\n{\n");
     fprintf(file,"\tif(!SDev)\tCloseDevice(&Sound);\n");
     fprintf(file,"\tif(SPort)\tDeletePort(SPort,sizeof(str
     uct MsgPort));\n");
     fprintf(file,"\tif(SData)\tFreeMem(SData,%d);\n}\n\n",
     ENDE);
     fprintf(file,"SHORT SoundOn()\n{\n");
     fprintf(file,"\tUBYTE *AllocMem();\n");
     fprintf(file,"\tCOUNT i;\n\n");
     fprintf(file,"\tif(!(SData=AllocMem(%d,MEMF_CHIP)))\tr
     eturn(1);\n",ENDE);
     fprintf(file,"\tfor(i=0;i<%d;i++)\n\t\tSData[i]=SD[i]
     ;\n",ENDE);
     fprintf(file,"\tif(!(SPort=CreatePort(\"SPort\",0))){\
     n");
     fprintf(file,"\t\tSoundOff();\n\t\treturn(1);\n\t}\n")
     ;
     fprintf(file,"\tSound.ioa_Request.io_Message.mn_ReplyP
     ort=SPort;\n");
     fprintf(file,"\tSound.ioa_Request.io_Message.mn_Node.l
     n_Pri=10;\n");
     fprintf(file,"\tSound.ioa_Data=&buffer;\n");
     fprintf(file,"\tSound.ioa_Length=(ULONG)sizeof(buffer)
     ;\n");
     fprintf(file,"\tif(SDev=OpenDevice(AUDIONAME,0,&Sound,
     0)){\n");
     fprintf(file,"\t\tSoundOff();\n\t\treturn(1);\n\t}\n")
     ;
     fprintf(file,"\tSound.ioa_Request.io_Command\t=CMD_WRI
     TE;\n");
     fprintf(file,"\tSound.ioa_Request.io_Flags\t=ADIOF_PER
     VOL|IOF_QUICK;\n");
     fprintf(file,"\tSound.ioa_Data\t\t\t=SData;\n");
     fprintf(file,"\tSound.ioa_Length\t\t=%d;\n",ENDE);
     fprintf(file,"\tSound.ioa_Period\t\t=(UWORD)%d;\n",p);
     fprintf(file,"\tSound.ioa_Volume\t\t=%d;\n",v);
     fprintf(file,"\tSound.ioa_Cycles\t\t=(UWORD)%d;\n",c);
     fprintf(file,"\treturn(0);\n}\n");
     fclose(file);
  }
  else DisplayBeep(0);
}
VOID CheckReq()                                /* Reques
ter kontrollieren */
{
    SHORT reqclear=0;
    while(reqclear==0){
        GetMessage();
        if(CLASS==GADGETUP){
            switch(GAD_ID){
                case 12:{
                    Save();
                    reqclear=1;
                    break;
                }
                case 13:{
                    reqclear=1;
                    break;
                }
                default:break;
            }
        }
    }
}
VOID main()
{
    SHORT x=0,y=0;
    BOOL ende=0;
    COUNT i,j;
    FLOAT k;
    openw();
    SetAPen(RP, 1);
    RectFill(RP,0,0,510,256);
    for(i=0;i<512;i+=2)    SetPoint(i,127);
    while(!ende){
        Wait(1<<win->UserPort->mp_SigBit);
        GetMessage();
        switch(CLASS){
            case MOUSEBUTTONS:{
                if(CODE==MENUDOWN){
                    Play();
                    while(CODE!=MENUUP) GetMessage();
                }
                else if(CODE==SELECTDOWN){
                    while(CODE!=SELECTUP){
                        GetMessage();
                        if((x!=MOUSX)||(y!=MOUSY)){
                            x=MOUSX;
                            y=MOUSY;
                            if(x<=510) SetPoint(2*(UBYTE)(
                            (FLOAT)x/2.0),y);
                        }
                    }
                }
                break;
            }
            case GADGETUP:{
                 switch(GAD_ID){
                     case 4:{    /* Null */
                         for(i=0;i<512;i+=2)    SetPoint(i
                         ,127);
                         break;
                     }
                     case 5:{    /* Sinus */
                         for(j=0;j<2;j++){
                             for(i=0;i<256;i+=2){
```

Listing 1. (Fortsetzung)

```
                        k=sqrt((FLOAT)  ((256-i)*(
                        256-i)+i*i));
                        k=j?-k+128.0:k-128.0;
                        SetPoint(i+256*j,(UBYTE)k)
                        ;
                    }
                }
                break;
            }
            case 6:{    /* 3Eck */
                for(j=0;j<2;j++){
                    for(i=j*256;i<128+j*256;i+=2)
                    {
                        k=(FLOAT)i*.7;
                        k=j?k+204.0:128.0-k;
                        SetPoint(i,(UBYTE)k);
                    }
                    for(i=128+j*256;i<256+j*256;i
                    +=2){
                        k=(FLOAT)i*.7;
                        k=j?-k-28:k-52.0;
                        SetPoint(i,(UBYTE)k);
                    }
                }
                break;
            }
            case 7:{    /* SZahn */
                for(j=0;j<2;j++){
                    for(i=j*256;i<128+j*256;i+=2)
                    {
                        k=.7*(FLOAT)(j*256+180-i);
                        SetPoint(i,(UBYTE)k);
                    }
                    for(i=128+j*256;i<256+j*256;i
                    +=2){
                        k=.7*(FLOAT)(j*256+436-i);
                        SetPoint(i,(UBYTE)k);
                    }
                }
                break;
            }
            case 8:{    /* Recheck */
                for(j=0;j<2;j++){
                    for(i=j*256;i<128+j*256;i+=2)
                        SetPoint(i,38);
                    for(i=128+j*256;i<256+j*256;i
                    +=2) SetPoint(i,218);
                }
                break;
            }
            case 9:{    /* 2mal */
                for(i=j=0;i<256;i+=2,j++) Data[j]
                =Data[i];
                for(i=128;i<256;i++)      Data[i]
                =Data[i-128];
                for(i=0;i<512;i+=2) SetPoint(i,(B
                YTE)(127-Data[i/2]));
                break;
            }
            case 10:{    /* Save */
                Request(&Req,win);
                CheckReq();
                break;
            }
            case 11:{
                ende=1;
                break;
            }
            default:break;
        }
        break;
    }
    case VANILLAKEY:{
        switch(CODE){
            case 49:{    /* Ende links */
                if((Ende>2)&&((Ende-2)>Start)){
                    Ende-=2;
                    SetPoint(Ende+2,(BYTE)(127-Dat
                    a[(Ende+2)/2]));
                    SetPoint(Ende,(BYTE)(127-Data[
                    Ende/2]));
                }
                break;
            }
            case 51:{    /* Ende rechts */
                if(Ende<510){
                    Ende+=2;
                    SetPoint(Ende-2,(BYTE)(127-Dat
                    a[(Ende-2)/2]));
                    SetPoint(Ende,(BYTE)(127-Data[
                    Ende/2]));
                }
                break;
            }
            case 52:{    /* Ende links schnell */
                if((Ende>8)&&((Ende-8)>Start)){
                    Ende-=8;
                    SetPoint(Ende+8,(BYTE)(127-Dat
                    a[(Ende+8)/2]));
                    SetPoint(Ende,(BYTE)(127-Data[
                    Ende/2]));
                }
                break;
            }
            case 54:{    /* Ende rechts schnell */
                if(Ende<506){
                    Ende+=8;
                    SetPoint(Ende-8,(BYTE)(127-Dat
                    a[(Ende-8)/2]));
                    SetPoint(Ende,(BYTE)(127-Data[
                    Ende/2]));
                }
                break;
            }
            case 50:
            case 53:{    /* Ende-Position */
                i=Ende;
                Ende=510;
                SetPoint(i,(BYTE)(127-Data[i/2]));
                SetPoint(510,(BYTE)(127-Data[255])
                );
                break;
            }
            case 55:{    /* Start links */
                if(Start>=8){
                    Start-=8;
                    SetPoint(Start+8,(BYTE)(127-Da
                    ta[(Start+8)/2]));
                    SetPoint(Start,(BYTE)(127-Data
                    [Start/2]));
                }
                break;
            }
            case 57:{    /* Start rechts */
                if((Start<504)&&((Start+8)<Ende)
                ){
                    Start+=8;
                    SetPoint(Start-8,(BYTE)(127-Da
                    ta[(Start-8)/2]));
                    SetPoint(Start,(BYTE)(127-Data
                    [Start/2]));
                }
                break;
            }
            case 56:{    /* Start-Position */
                i=Start;
                Start=0;
                SetPoint(i,(BYTE)(127-Data[i/2]));
                SetPoint(0,(BYTE)(127-Data[0]));
                break;
            }
            default:break;
        }
        break;
    }
    default:break;
  }
 }
 closew();
}
(C) 1989 M&T
```

Listing 1. (Schluß)

```
#include <devices/audio.h>  /* notwendige Headerfiles einbinden */
#include <exec/memory.h>
#define Play(); BeginIO(&Sound); WaitIO(&Sound);                /*
  Abspielen des Sounds */

UBYTE buffer=15,*SData,SD[56]=
{                     /* globale Variablen und Sounddaten (hier 56)*/
  1,  7, 13, 18, 24, 29, 35, 41, 46, 52, 57, 63, 69, 74, 80, 85,
168,173,179,185,190,196,201,207,213,218,224,229,235,241,246,252,
  1,  7, 13, 18, 24, 29, 35, 41, 46, 52, 57, 63, 69, 74, 80, 85,
168,173,179,185,190,196,201,207
};
extern struct MsgPort   *CreatePort();
extern struct Device    *OpenDevice();
struct MsgPort          *SPort;
struct Device           *SDev;
struct IOAudio           Sound;

VOID SoundOff()                           /* Resourcen freigeben */
{
      if(!SDev)   CloseDevice(&Sound);    /* Sounddevice schließen */
      if(SPort)   DeletePort(SPort,sizeof(struct MsgPort));
                  /* Port */
      if(SData)   FreeMem(SData,56);  /* Speicherbereich
                  freigeben */
}

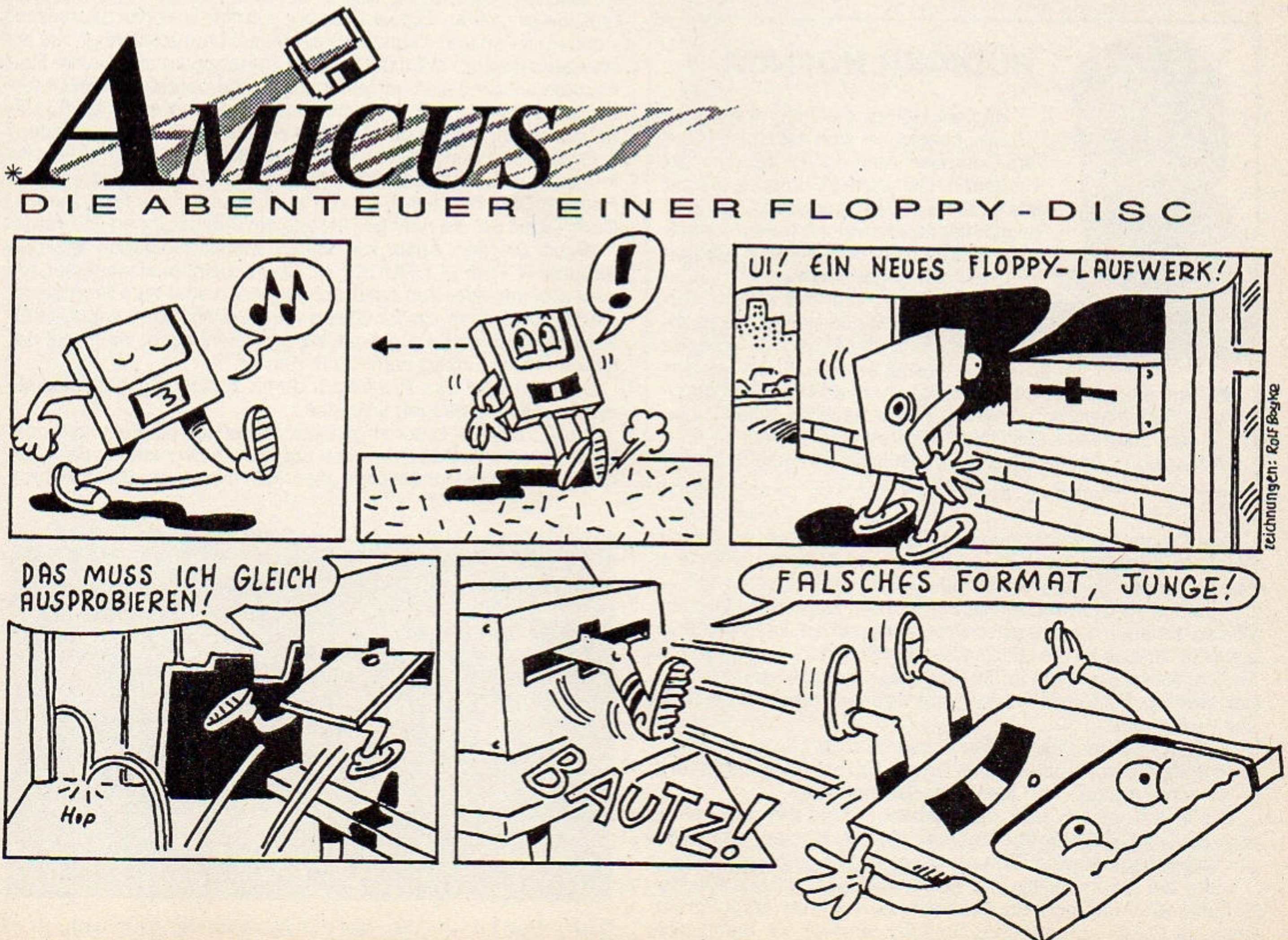
SHORT SoundOn()     /* Klang initialisieren; Rückgabewert im
                                          Fehlerfalle: 1 */
{
        UBYTE *AllocMem();
        COUNT i;

/* Mit AllocMem() Speicher für Daten im Chipmemory anfordern: */

        if(!(SData=AllocMem(56,MEMF_CHIP)))      return(1);

for(i=0;i<56;i++)
        SData[i]=SD[i]; /* Daten ins Chipmemory kopieren */
if(!(SPort=CreatePort("SPort",0)))
{                                   /* Antwortport kreieren */
        SoundOff();
        return(1);
}
Sound.ioa_Request.io_Message.mn_ReplyPort=SPort;
                                        /* Portzuweisung */
Sound.ioa_Request.io_Message.mn_Node.ln_Pri=10;
                                       /* Prozessespriorität*/
Sound.ioa_Data=&buffer; /* Soundkanäle festlegen */
Sound.ioa_Length=(ULONG)sizeof(buffer);
                                       /* Größe des Buffers */
if(SDev=OpenDevice(AUDIONAME,0,&Sound,0))
{
                                      /* Audio-Device öffnen */
SoundOff();                   /* hat nicht geklappt */
return(1);                    /* daher zurück mit 1 */
}
Sound.ioa_Request.io_Command     =CMD_WRITE;
                                     /* Ausgabe vorbereiten */
Sound.ioa_Request.io_Flags       =ADIOF_PERVOL| IOF_QUICK;
                                         /* Flags */
Sound.ioa_Data                   =SData;
                                /* Zeiger auf die Sounddaten */
Sound.ioa_Length                 =56;
                                /* Anzahl der Werte (UBYTE) */
Sound.ioa_Period            =(UWORD)255;  /* Samplerperiode */
Sound.ioa_Volume                 =64;          /* Lautstärke */
Sound.ioa_Cycles           =(UWORD)1;     /* Wiederholungen */
return(0);                               /* Null: alles OK */
}
```

Listing 2. Ein dokumentiertes Listing zum Quellcode einer Sounddatei, die vom SEEd erzeugt wurde



ETIKETTEN

Es ist gut, viele Disketten mit vielen Programmen zu haben. »DiskEti« erleichtert Ihnen durch Etiketten die Suche in Ihren Beständen. Komfortable Bedienung ist dabei Trumpf.



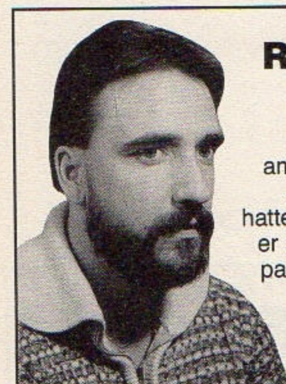
Meist steht auf den Diskettenaufklebern nur ein Name. Werden es aber mehr und mehr Disketten, verliert man den Überblick. Die Zeit, die man für das Suchen verschwendet, wird immer länger. Etiketten für 3-½-Zoll-Disketten mit den Namen der wichtigsten Dateien helfen das Gesuchte schneller zu finden.

Um diese Etiketten zu drucken, gibt es verschiedene Wege. Einen wirklich komfortablen geht »DiskEti«. Das Einlesen der Dateinamen geschieht durch das Programm. Somit entfällt schon viel Tipparbeit. Welche Namen auf das Etikett gedruckt werden, entscheiden Sie durch Anklicken mit der linken Maustaste. Die Namen können auf Wunsch editiert werden. All diese Gründe sprechen für die Wahl von DiskEti zum Programm des Monats.

Für manche Routinen braucht Amiga-Basic die Unterstützung des Betriebssystems. Damit das funktioniert, benötigt DiskEti die vier Dateien:

☐ dos.bmap ☐ exec.bmap
☐ graphics.bmap ☐ intuition.bmap

Sie finden diese Dateien auf Ihrer Extras-Diskette. Nicht vorhandene ».bmap«-Dateien generieren Sie wie im Artikel »Neue Dimensionen in Basic«, Ausgabe 1/89, Seite 147, beschrieben. Kopieren Sie die Dateien in dasselbe Verzeichnis, in dem DiskEti und Amiga-Basic stehen. Nach dem Start erscheint zunächst eine Kurzanleitung zum Programm. Nach einem Klick mit der linken Maustaste sehen Sie die Arbeitsfläche des Programms (Bild 1). In dem Fenster sind alle benötigten Schalter zu sehen. Es werden keine Pull-Down-Menüs verwendet.



RÜDIGER HOFNER

Rüdiger Hofner, der Autor von »DiskEti«, bezeichnet sich selbst als »Opa am Computer« und das im Alter von 30 Jahren. Den ersten Computerkontakt hatte er Ende 1984 mit einem VC 20, den er sich kaufte, um seinen Kindern in ein paar Jahren nicht unwissend in Sachen Computer gegenüberstehen zu müssen. Aus dem »nur mal sehen« wurde ein richtiges Hobby. Als nächster Computer folgte, wie sollte es anders sein, ein C 64. Im Jahr 1987 war es dann soweit, ein Amiga kam ins Haus. Sein nächster Schritt wird das Lernen der Programmiersprache C sein. Von dem 2000-Mark-Gewinn (Programm des Monats im AMIGA-Magazin 2/89) kaufte er eine Festplatte und gute Software.

Das Programm wartet nun auf die Auswahl eines Laufwerks. DiskEti unterstützt die Laufwerke DF0: und DF1:. Klicken Sie einen der beiden Schalter (»df0:« oder »df1:«) an. Im oberen Textfeld am rechten Fensterrand sehen Sie nun den Namen der Diskette. Wie später auch die einzelnen Programmnamen, kann er editiert werden. Um das zu tun, klicken Sie den Schalter »Name ändern« an. Der Name erscheint im Textfeld darunter. Dort kann er verändert werden. Folgende Tastenkombinationen stehen Ihnen dabei zur Verfügung:

— <Shift - Cursor links>: Der Cursor springt an den Anfang.
— <Shift - Cursor rechts>: Der Cursor wird an das Ende gesetzt.
— <rechte Amiga - x>: löscht die Zeichenkette.

Die Tasten <Del> und <Backspace> sind wie gewohnt belegt. Mit <RETURN> übernehmen Sie den Namen. Wollen Sie ihn verbessern, klicken Sie erneut auf »Name ändern«.

Sieht die Zeichenkette wie gewünscht aus, klicken Sie den Schalter »übernehmen« an. Der Name wird dann in das unten dargestellte Etikett eingetragen. Natürlich ändern Sie damit nicht den Namen der Datei auf Diskette, sondern nur den Namen für den Ausdruck.

Mit dem Schalter »nächste Datei« übergehen Sie Dateien, die nicht auf den Aufkleber gedruckt werden sollen. Genauso ist »nächstes Dir« zu verwenden. Es läßt ein gesamtes Unterverzeichnis aus. Das ist sinnvoll, wenn es sich zum Beispiel um ständig wiederkehrende Verzeichnisse wie LIBS: handelt.

Nachdem Sie nun den Diskettennamen festgelegt haben, folgen die einzelnen Dateien. Dabei werden Unterverzeichnisse, ».info«-Dateien und ».fastdir«-Dateien nicht berücksichtigt. Als erstes zeigt Ihnen DiskEti die Dateien im Hauptverzeichnis der Reihe nach an. Sind sie bearbeitet, folgen die Dateien in den Unterverzeichnissen. Pro Subdirectory kann das Programm 100 Einträge verwalten, die vor der Anzeige alphabetisch sortiert werden.

Sollten Sie beim Arbeiten merken, daß sich die falsche Diskette im Laufwerk befindet, ist das kein Problem. Durch Aktivieren von »andere Diskette« kehrt DiskEti zum Programmanfang zurück. Dann legen Sie die richtige Diskette ein und beginnen von vorne.

Wenn Sie den Ausdruck starten wollen, betätigen Sie den Schalter »ETIKETT DRUCKEN«. Es erscheint ein Requester, mit dem Sie entweder den Ausdruck beginnen oder zum Programmstart zurückgehen. Ist das Etikett voll (32 Einträge), bringt DiskEti automatisch den Requester. Dies geschieht auch, wenn auf der Diskette kein Eintrag mehr zu finden ist.

Beendet wird das Programm durch Betätigung des Schließsymbols (close gadget) links oben.

DiskEti arbeitet in der abgedruckten Version mit Epson-kompatiblen Druckern. Das Umstellen auf andere Drucker ist, dank der sauberen Programmierung, aber kein Problem. Die für Ihren

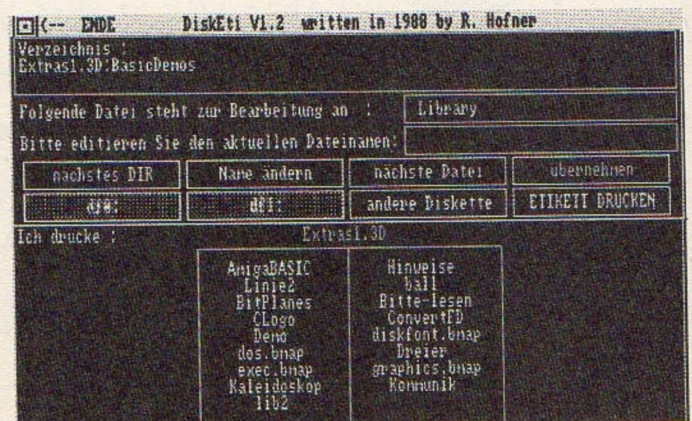


Bild 1. Das ist »DiskEti« bei der Auswahl der Dateinamen

MIT PFIFF

Drucker gültigen Sequenzen finden Sie im Handbuch. Folgende Zeichenketten werden verwendet, um den Drucker anzusteuern:
— esc$: das Escape-Zeichen, das Steuersequenzen einleitet;
— init$: Steuersequenz zum Initialisieren des Druckers;
— fett$: Einschalten des Fettdrucks;
— abstand$: dient zur Festlegung des Zeilenabstands in 1/72 Zoll. Hier ist bei einigen Druckern eine zweite Sequenz nötig, um den eingestellten Abstand zu aktivieren. Beim Star NL-10 funktioniert dies so:

```
PRINT #1,esc$;"A";CHR$(6)
PRINT #1,esc$;"2"
```

— klein$: Stellt die Schriftart Superscript (hochgestellt) ein. In Verbindung mit dem geänderten Zeilenabstand passen so mehr Daten auf das Etikett.
— eng$: Druck in Schmalschrift (132 Zeichen pro Zeile);
— elite$: die Sequenz, die den Drucker anweist, in der Schriftart Elite zu drucken (12 Zeichen pro Zoll);
— Ende$: stellt wieder den normalen Zeilenabstand her.

Die Festlegung dieser Zeichenketten geschieht in dem Unterprogramm »Etikett« ab Zeile 627. Die entsprechenden Sequenzen entnehmen Sie Ihrem Druckerhandbuch. Doch dies ist wie gesagt nur notwendig, wenn Ihr Drucker mit der vorliegenden Version nicht richtig arbeitet.

Die Endlosetiketten für 3½-Zoll-Disketten sind in jedem gut sortierten Schreibwarenhandel erhältlich. *R. Hofner/R. Beaupoil/kn*

LabelMaker

AmigaBasic	Etikett-Ed
AddBuffers	Assign
BindDrivers	Break
CD	ChangeTaskPri
Copy	Date
Delete	Dir
DiskChange	DiskDoctor
Echo	Ed
Edit	Else
EndCLI	EndIf
Execute	FailAt
Fault	FileNote
If	Info
Install	Join
Lab	List
LoadWb	MakeDir

Bild 2. Ein Ausdruck von »DiskEti« mit einem Epson-Drucker

Programmname:	DiskEti
Computer:	A500, A1000, A2000 mit Kickstart 1.2 & 1.3
Sprache:	Amiga-Basic 1.2

```
Programmautor: Rüdiger Hofner
------------------------------
 1 rHO CLEAR,30000
 2 eG  DEFLNG a-z
 3 tv  REM ###
 4 iU  SCREEN 1,640,256,2,2
 5 kj  PALETTE 0,0,0,1
 6 lD  PALETTE 1,.9,.9,.9
 7 qv  PALETTE 2,0,0,0
 8 JW  LIBRARY "intuition.library"
 9 39  LIBRARY "graphics.library"
10 Hw  WINDOW CLOSE 1
11 j5  WINDOW 2,,,0,1
12 nz  Fenster&=WINDOW(7)
13 ey  POKEW Fenster&+98,0
14 81  CALL RefreshWindowFrame&(Fenster&)
15 Hm  Meldung:
16 op1  ue$=CHR$(252)
17 JK   LOCATE 3,18
18 RJ   Schatten "Herzlich willkommen zu DiskEti",0
19 fI   PRINT
20 D1   Schatten "Vorweg erst mal ein paar kleine Anmerkungen:",0
21 Uv   Schatten "Ich lese die Verzeichnisse einer",1
22 4p   Schatten " von Ihnen bestimmten Diskette",0
23 cV   Schatten "Dabei ist folgendes zu beachten:",0
24 kN   PRINT
25 GS   Schatten "- Ich lese maximal 100 Files pro Verzeichnis",0
26 MZ   Schatten "- Ich zeige keine Directory-Namen in der Auflist
        ung an",0
27 e5   Schatten "- Ich zeige keine .info-Dateien in der Auflistun
        g an",0
28 rQ   Schatten "- Die Namen sind maximal 15 Zeichen lang",0
29 pS   PRINT
30 TM   Schatten "Die Dateien werden alphabetisch sortiert und ein
        zeln",1
31 i4   Schatten " angezeigt. Dann können ",0
32 YR   Schatten "Sie diesen Dateinamen editieren, "+ue$+"bersprin
        gen,",1
33 wH   Schatten " zum Druck "+ue$+"bernehmen",0
34 7s   Schatten "oder aber direkt in das nächste Unterverzeichnis
        ",1
35 ep   Schatten "wechseln. Das Wechseln ",0
36 7T   Schatten "passiert automatisch am Ende eines Verzeichnisse
        s. ",1
37 7c   Schatten "Von Ihnen "+ue$+"bernommene",0
38 QM   Schatten "Dateien werden ständig angezeigt!",0
39 1x   Schatten "Der Name der Diskette wird auch zum Druck "+ue$+
        "bernommen!!",0
40 1A   Schatten "(Den können Sie aber auch ändern!!)",0
41 1e   PRINT
42 oN   Schatten "ACHTUNG!!!!!    Ich "+ue$+"bernehme maximal 32 D
        ateinamen",0
43 3g   PRINT
44 hA   Schatten ".....linke Maustaste dr"+ue$+"cken.....",0
45 Vz   Maus
46 IO   CLS
47 60   LOCATE 12,29
48 xM   Schatten "Einen Moment bitte !",0
49 DV   LOCATE 14,18
50 ax   Schatten "Ich muß erst mal 'n paar .bmaps lesen.......",0
51 ONO LIBRARY "dos.library"
52 qT  LIBRARY "exec.library"
53 iU  DECLARE FUNCTION ExNext% LIBRARY
54 sQ  DECLARE FUNCTION Lock LIBRARY
55 9c  DECLARE FUNCTION AllocMem LIBRARY
56 el  DECLARE FUNCTION IoErr% LIBRARY
57 01  DECLARE FUNCTION GetMsg LIBRARY
58 kj  DECLARE FUNCTION OpenWindow LIBRARY
59 1R  DECLARE FUNCTION Examine% LIBRARY
60 26  DECLARE FUNCTION ViewAddress LIBRARY
61 rb  DECLARE FUNCTION ViewPortAddress LIBRARY
62 AJ  DECLARE FUNCTION AutoRequest LIBRARY
63 TC  variablen:
64 qd1  zaehler% = 1
65 bu   SpeicherListe  = 0&
66 5m   i%     = 0
67 yM   x%     = 100
68 1Q   y%     = 100
69 bh   lo1%   = 3
70 m4   DIM SHARED dir.Name$(x%)
71 ay   DIM SHARED dir.Typ$(x%)
72 PZ   DIM SHARED info$(x%), Druck$(x%)
73 3x   DIM a$(y%)
74 84   DIM a&(y%)
```

Listing. Mit »DiskEti« bedrucken Sie Etikettenaufkleber schnell und komfortabel. Bitte mit dem Checksummer (Seite 143) eingeben.

```
75 VGO Fenster:
76 e11  MacheRand FeldRand,-1,-1,1,150,14
77 VN   MacheRand AusgRand,-1,-1,3,630,28
78 OF   MacheRand AusgRand1,-1,-1,3,260,14
79 NL   IntText EditTxt,1,1,31,2,"Name ändern",0&
80 VQ   IntText WeiterTxt,1,1,23,2,"nächste Datei",0&
81 cV   IntText OkTxt,3,1,35,2,"Übernehmen",0&
82 t0   IntText GoonTxt,1,1,27,2,"nächstes DIR",0&
83 vt   IntText df0Txt,1,1,59,2,"df0:",0&
84 22   IntText df1Txt,1,1,59,2,"df1:",0&
85 yF   IntText DiskTxt,1,1,15,2,"andere Diskette",0&
86 KH   IntText DruckTxt,1,1,15,2,"ETIKETT DRUCKEN",0&
87 Wz   IntText FrageTxt,1,2,10,8,"Wollen Sie wirklich aufhören ??
     ",0&
88 Z7   IntText JaTxt,1,2,6,3,"Ja !!",0&
89 fT   IntText NeinTxt,1,2,6,3,"Nein",0&
90 OR   IntText DruckFrageTxt,1,2,10,8,SPACE$(11)+"Drucken ??",0&
91 DJ   FeldDef GoonFeld,0&,12,70,148,12,0,1,1,FeldRand,GoonTxt,0&
     ,1
92 6c   FeldDef EditFeld,GoonFeld,167,70,148,12,0,1,1,FeldRand,Edi
     Txt,0&,2
93 zn   FeldDef WeiterFeld,EditFeld,322,70,148,12,0,1,1,FeldRand,W
     eiterTxt,0&,3
94 y2   FeldDef OkFeld,WeiterFeld,477,70,148,12,0,1,1,FeldRand,OkT
     xt,0&,4
95 Yr   FeldDef df0Feld,OkFeld,12,85,148,12,0,1,1,FeldRand,df0Txt,
     0&,5
96 d3   FeldDef df1Feld,df0Feld,167,85,148,12,0,1,1,FeldRand,df1Tx
     t,0&,6
97 aA   FeldDef DiskFeld,df1Feld,322,85,148,12,0,1,1,FeldRand,Disk
     Txt,0&,7
98 1m   FeldDef DruckFeld,DiskFeld,477,85,148,12,0,1,1,FeldRand,Dr
     uckTxt,0&,8
99 Ic   titel$="<-- ENDE"
100 P8  titel$=titel$+SPACE$(8)+"DiskEti V1.2 "
101 hw  titel$=titel$+" written in 1988 by R. Hofner"
102 qW  WINDOW CLOSE 2
103 1o  WINDOW 3,"",(0,99)-(629,242),0,1
104 WY0 REM ###
105 Co1 FensterDef NFenster,0,0,639,100,64+512&,8&+4096&,DruckFeld
     ,titel$
106 KD  FensterAdresse=OpenWindow(NFenster)
107 np  IF FensterAdresse=0 THEN ERROR 7
108 09  FensterRP=PEEKL(FensterAdresse+50)
109 k5  FensterUP=PEEKL(FensterAdresse+86)
110 do  DrawBorder FensterRP,AusgRand,5,11
111 TG  DrawBorder FensterRP,AusgRand1,375,40
112 le  DrawBorder FensterRP,AusgRand1,375,55
113 SU  CALL SetAPen(FensterRP,1) 'Schriftfarbe f. Druck1 auf norm
     ale Vordergr.F.
114 qb  druck1 "Folgende Datei steht zur Bearbeitung an  :",9,41
115 xD  druck1 "Bitte editieren Sie den aktuellen Dateinamen:",9,5
     6
116 to  GOSUB ZeichneEtikett
117 aD0 Hauptprogramm:
118 Jv1  GOSUB FelderEin
119 um   Fuelle 6,11,631,36
120 fz   druck1 "Bitte wählen Sie das Laufwerk aus ...",9,19
121 Hn   GOSUB Meldungsschleife
122 xp   Fuelle 6,11,631,36
123 ZV0 LiesDir:
124 Fd1  FOR schleife% = jetzt% TO zaehler%-1
125 CA4    IF a&(schleife%) = stufe% THEN
126 BZ6      such$=a$(schleife%)
127 c7       z% = x%
128 1G       zeige%=0
129 4w       Fuelle 6,11,631,36
130 nP       druck1 "Ich lese : ",9,11:druck1 a$(schleife%),9,19
131 Sc       druck1 "Bitte ein wenig Geduld......",9,27
132 eR       LiesDisk such$, x%
133 A1       IF fehler%=500 THEN
134 GU7       neu%=1
135 Yc        GOTO nochmal
136 816      END IF
137 wh       max% = x%
138 nI       x% = z%
139 mY       Sort
140 3F       IF schleife%=0 THEN
141 KK8        a$(0)=dir.Name$(0)+":"
142 Yq         Druck$(0)=LEFT$(a$(0),LEN(a$(0))-1)
143 IA         Fuelle 6,11,631,36
144 U1         druck1 "Diskettenname:",9,19
145 6Q         IF LEN(Druck$(0)) > 15 THEN
146 A1A          Druck$(0)=LEFT$(Druck$(0),15)
147 JC8        END IF
148 Su         druck1 Druck$(0),388,41
149 8o         ed$=Druck$(0)
150 kA         GOSUB Meldungsschleife
151 Cz         IF neu% = 1 THEN GOTO nochmal
152 OH6      END IF
153 3n       z$ = a$(schleife%)
154 LC       dir$ = LEFT$(z$,LEN(z$)-1)
155 UM       Fuelle 6,11,631,36
156 Pz       druck1 "Verzeichnis :",9,11:druck1 dir$,9,19
157 cz       FOR anz% = 1 TO max%
158 bHB         IF dir.Typ$(anz%) <> "DIR" THEN
159 9JD           IF RIGHT$(dir.Name$(anz%),5) <> ".info" THEN
160 SqF             IF RIGHT$(dir.Name$(anz%),8) <> ".fastdir"
                     THEN
161 xoH               info$(zeige%)=dir.Name$(anz%)
162 Si                zeige%=zeige%+1
163 ZSF             END IF
164 aTD           END IF
165 GzB         ELSE
166 rxD           f1=1
167 pS            a$(zaehler%)=a$(schleife%)+dir.Name$(anz%)+"/"
168 At            a&(zaehler%)=stufe%+1
169 oo            zaehler%=zaehler%+1
170 gZB         END IF
171 vq6      NEXT anz%
172 ib4    END IF
173 3S     ausgabe%=0
174 nK     Abfrage:
175 MP5     IF ausgabe% < zeige% THEN
176 e47       info$(ausgabe%)=LEFT$(info$(ausgabe%),15)
177 Op        Fuelle 380,40,631,50
178 mX        druck1 info$(ausgabe%),388,41
179 Gc        ed$=info$(ausgabe%)
180 Ee        GOSUB Meldungsschleife
181 Wp        IF neu% = 1 OR drucken% = 1 THEN GOTO nochmal
182 Sz        ausgabe%=ausgabe%+1
183 w2        GOTO Abfrage
184 un5     END IF
185 2y1  NEXT schleife%
186 hm   jetzt% = schleife%
187 QG   IF f1=1 THEN
188 8D4    f1=0
189 3W     stufe% =stufe%+1
190 e9     GOTO LiesDir
191 1u1  END IF
192 Ty0 nochmal:
193 Jp1  IF neu% = 0 THEN
194 bC3    Etikett
195 5y1  END IF
196 J8   Fuelle 380,40,631,50
197 is   Fuelle 9,84,317,97
198 Ct   GOSUB neuinit
199 Ug   GOTO Hauptprogramm
200 6V0 AllesAus:
201 re1  CALL CloseWindow(FensterAdresse)
202 QE   CALL SpeicherFrei
203 YF   WINDOW CLOSE 3
204 8A0 REM ###
205 5A1  SCREEN CLOSE 1
206 yC   LIBRARY CLOSE
207 50   END
208 qy0 Meldungsschleife:
209 Jz1  Lauf%=1
210 Yg   Hauptschleife:
211 rT   IF Lauf% =1 THEN
212 Ow3    IntuiMeldung = GetMsg(FensterUP)
213 we     IF IntuiMeldung > 0 THEN GOSUB IntuitionMeldung
214 Ct     GOTO Hauptschleife
215 PI1  END IF
216 qS0 RETURN
217 Ab  IntuitionMeldung:
218 mc1  MeldungsTyp=PEEKL(IntuiMeldung+20)
219 ei   Platz=PEEKL(IntuiMeldung+28)
220 76   FeldNr%=PEEK(Platz+39)
221 DN   CALL ReplyMsg&(IntuiMeldung)
222 Cq   IF (MeldungsTyp=64&) THEN
223 nr3    IF FeldNr% = 1 THEN
224 VA6      Lauf%=0
225 RI       ausgabe%=zeige%
226 6F3    ELSEIF FeldNr% = 2 THEN
227 VL6      StrInfo EString,ed$
```

```
ZE        FeldDef StrGad,0&,389,57,150,12,0,1,4,0&,0&,EString,9
OC        CALL AddGadget(FensterAdresse,StrGad,-1)
u5        CALL RefreshGadgets(StrGad,FensterAdresse,0&)
1X        CALL ActivateGadget(StrGad,FensterAdresse,0&)
GQ3     ELSEIF FeldNr% = 3 THEN
eJ6       Lauf%=0
MX3     ELSEIF FeldNr% = 4 THEN
6m6       IF Druck% > 0 THEN
Hf8         Druck$(Druck%)=info$(ausgabe%)
1e6       END IF
WT        GOSUB PrintOut
VR        Druck%=Druck%+1
1Q        Lauf%=0
ms        IF Druck% = 33 THEN drucken% = 1
Yk3     ELSEIF FeldNr% = 5 THEN
oT6       Lauf%=0
SW        a$(0)="df0:"
Yy        GOSUB FelderAus
gt3     ELSEIF FeldNr% = 6 THEN
sX6       Lauf%=0
cc        a$(0)="df1:"
c2        GOSUB FelderAus
o23     ELSEIF FeldNr% = 7 THEN
p06       neu% = 1
wb        Lauf% = 0
vA3     ELSEIF FeldNr% = 8 THEN
ze6       Lauf%=0
NG        drucken% = 1
2I3     ELSEIF FeldNr% = 9 THEN
T26       LiesEdit EString,r$
T8        IF Druck% < 1 THEN
N08         Druck$(0)=r$
LA          Fuelle 380,40,631,50
Hj          druck1 Druck$(0),388,41
yU          CALL RemoveGadget(FensterAdresse,StrGad)
Eg          CALL RefreshGadgets(GoonFeld,FensterAdresse,0&)
ra6       ELSE
Jm8         info$(ausgabe%)=r$
RG          Fuelle 380,40,631,50
Dy          druck1 info$(ausgabe%),388,41
4a          CALL RemoveGadget(FensterAdresse,StrGad)
Km          CALL RefreshGadgets(GoonFeld,FensterAdresse,0&)
IB6       END IF
3C        Fuelle 376,55,631,66
KD3     END IF
LE1   END IF
K6    IF MeldungsTyp=512& THEN
D13     Antwort=AutoRequest(FensterAdresse,FrageTxt,JaTxt,NeinTx
        t,0&,0&,300,50)
Uj      IF Antwort > 0 THEN GOTO AllesAus
PI1   END IF
qS0  RETURN
IJ   PrintOut:
nS1   IF Druck% = 0 THEN
123     lo%=0
Pj      COLOR 3,0
nJ      LOCATE 1,15
vB      v=INT((50-LEN(Druck$(Druck%)))/2)
RM      PRINT SPC(v);Druck$(Druck%);
Nf      COLOR 1,0
Ex1   ELSE
KB3     IF lo%=1 THEN
u05       LOCATE lo1%,42
H03     ELSE
y45       LOCATE lo1%,24
eX3     END IF
DU      v=INT((15-LEN(Druck$(Druck%)))/2)
Yk      n=INT((15-LEN(Druck$(Druck%)))/2+.5)
sc      PRINT SPC(v);Druck$(Druck%);SPC(n);CHR$(32);
wt      lo%=lo%+1
XP      IF lo%=2 THEN
IJ4      lo%=0
nC       lo1%=lo1%+1
mf3     END IF
ng1   END IF
Eq0  RETURN
1b   neuinit:
cA1   neu%      = 0
Ke    Druck%    = 0
cA    stufe%    = 0
AI    zaehler%  = 1
d8    jetzt%    = 0
YX    i%        = 0
wW    schleife% = 0
xa    lo1%      = 3
dq    drucken%  = 0
4z    GOSUB ZeichneEtikett
Q20  RETURN
fw   FelderEin:
OM1   CALL OnGadget(df0Feld,FensterAdresse,0&)
SR    CALL OnGadget(df1Feld,FensterAdresse,0&)
K4    CALL OffGadget(EditFeld,FensterAdresse,0&)
ET    CALL OffGadget(WeiterFeld,FensterAdresse,0&)
9g    CALL OffGadget(OkFeld,FensterAdresse,0&)
sp    CALL OffGadget(GoonFeld,FensterAdresse,0&)
WL    CALL OffGadget(DiskFeld,FensterAdresse,0&)
4K    CALL OffGadget(DruckFeld,FensterAdresse,0&)
Ra    CALL RefreshGadgets(df1Feld,FensterAdresse,0&)
bD0  RETURN
pJ   FelderAus:
811   CALL OffGadget(df0Feld,FensterAdresse,0&)
Dr    CALL OffGadget(df1Feld,FensterAdresse,0&)
7u    CALL OnGadget(EditFeld,FensterAdresse,0&)
ko    CALL OnGadget(WeiterFeld,FensterAdresse,0&)
Jj    CALL OnGadget(OkFeld,FensterAdresse,0&)
cS    CALL OnGadget(GoonFeld,FensterAdresse,0&)
96    CALL OnGadget(DiskFeld,FensterAdresse,0&)
4e    CALL OnGadget(DruckFeld,FensterAdresse,0&)
S9    Fuelle 4,69,632,97
ZK    CALL RefreshGadgets(DruckFeld,FensterAdresse,0&)
nP0  RETURN
Iv   ZeichneEtikett:
171   CLS
Yw    LINE (171,10)-(461,151),3,b
5D    LINE (172,10)-(172,151),3
sw    LINE (314,10)-(314,151),3
28    LINE (315,10)-(315,151),3
y6    LINE (460,10)-(460,151),3
ss    PRINT "Ich drucke : "
wY0  RETURN
2Q   Ende:
g7   '**********************************************************
     ****
Z9   '**               Jetzt kommen die Sub's !!
       **
19   '**********************************************************
     ****
uz   REM Speicherbereiche reservieren:
X4   '------------------------------
1x   SUB ReservSpeicher(Puffer, Groesse) STATIC
791   SHARED SpeicherListe
kA    Groesse=Groesse+8
4U    Puffer=AllocMem(Groesse,65538&)
nD    IF Puffer>0 THEN
173     POKEL Puffer,SpeicherListe
To      POKEL Puffer+4,Groesse
2L      SpeicherListe=Puffer
C1      Puffer=Puffer+8
RA1   ELSE
RY3     ERROR 7
ch1   END IF
vx0  END SUB
TO   REM Speicherbereiche wieder freigeben:
yV   '------------------------------------
Ao   SUB SpeicherFrei STATIC
M01   SHARED SpeicherListe
vU    freimachen:
fI3     IF SpeicherListe>0 THEN
rJ5       Adresse=PEEKL(SpeicherListe)
Af        Groesse=PEEKL(SpeicherListe+4)
mW        FreeMem SpeicherListe, Groesse
NH        SpeicherListe = Adresse
rz        GOTO freimachen
1u3     END IF
8A0  END SUB
1k   REM NewWindow Struktur anlegen:
Wd   '----------------------------
```



Listing. »DiskEti« (Fortsetzung)

```
mK  SUB FensterDef(bs, x%, y%, b%, h%, IDCMP, f, Feld, T$) STAT
    IC
br1  Groesse = 48+LEN(T$)+1  'benötigte Speichergröße f. Fenste
    rstruktur
LY   ReservSpeicher bs,Groesse
ax   POKEW bs,x%          'xKoordinate linke obere Ecke
fB   POKEW bs+2,y%        'yKoordinate linke obere Ecke
6B   POKEW bs+4,b%        'Breite
XG   POKEW bs+6,h%        'Hoehe
6V   POKEW bs+8,65535&    'Fein- und Grobstift
5P   POKEL bs+10,IDCMP    'Nachrichten-Flags
mf   POKEL bs+14,f        'Fenster-Flags
Qr   POKEL bs+18,Feld     'erstes Gadget
4T   POKEL bs+26,bs+48    'Titel
BD0 REM # # #
XZ  WINDOW 3
pt  POKEL bs+30,PEEKL(WINDOW(7)+46)  'Zeiger auf Basic-Screen
vn1  POKEW bs+46,15         'Screen-Typ
Re   FOR i%=1 TO LEN(T$)
Y53    POKE bs+47+i%,ASC(MID$(T$,i%,1))
Wb1  NEXT
UW0 END SUB
P8  REM Gadget Struktur anlegen:
9g  '----------------------------
ds  SUB FeldDef(bs,nx,x%,y%,b%,h%,f%,a%,T%,i,txt,si,n%) STATIC
001  ReservSpeicher bs,44&        'Laenge der Struktur
5k   POKEL bs,nx          'naechstes Gadget
5R   POKEW bs+4,x%        'xKoordinate linke obere Ecke
Pf   POKEW bs+6,y%        'yKoordinate linke obere Ecke
qf   POKEW bs+8,b%        'Breite
Mf   POKEW bs+10,h%       'Hoehe
Oz   POKEW bs+12,f%       'Gadget-Flags
u5   POKEW bs+14,a%       'Aktivierung
E3   POKEW bs+16,T%       'Gadget-Typ
J9   POKEL bs+18,i        'Bild oder Rand
7A   POKEL bs+26,txt      'Text
a2   POKEL bs+34,si       'Zeiger auf Info-Struktur
qK   POKEW bs+38,n%       'Gadget-Nummer
1n0 END SUB
KH  REM IntuitionText Struktur anlegen:
vp  '---------------------------------
d4  SUB IntText(bs,c1%,d1%,x%,y%,T$,nx) STATIC
8E1  Groesse=20+LEN(T$)+1    'Strukturlaenge+Textlaenge+Nullbyt
    e
yB   ReservSpeicher bs,Groesse
fX   POKE bs,c1%          'Schriftfarbe
iF   POKE bs+2,d1%        'Schreibmodus
4w   POKEW bs+4,x%        'xKoordinate Textanfang
OA   POKEW bs+6,y%        'yKoordinate Textanfang
pl   POKEL bs+12,bs+20    'Text
ms   POKEL bs+16,nx       'naechster Text
xA   FOR i%=1 TO LEN(T$)
1N3    POKE bs+19+i%,ASC(MID$(T$,i%,1))
271  NEXT
020 END SUB
fN  REM Raender:
FZ  '-----------
mw  SUB MacheRand(bs,x%,y%,C%,b%,h%) STATIC
nu1  ReservSpeicher bs,48&         'Strukturlaenge+Koordinatenta
    belle
im   POKEW bs,x%            'xKoordinate
o1   POKEW bs+2,y%          'yKoordinate
WK   POKE bs+4,C%           'Zeichenfarbe
mH   POKE bs+7,8            '
Vr   POKEL bs+8,bs+16       'xy-Koordinatentabelle
A7   FOR i%=0 TO 1
VT3    POKEW bs+22+i%*4,h%-1
C6     POKEW bs+24+i%*4,b%-1
sN     POKEW bs+32+i%*4,1
qq     POKEW bs+38+i%*4,h%-1
L4     POKEW bs+40+i%*4,b%-2
JO1  NEXT
HJO END SUB
CG  REM Files alphab. sortieren:
wT  '--------------------------
p2  SUB Sort STATIC
gN2   SHARED max%
JB    Fuelle 6,11,631,36
QN    druck1 "Ich sortiere ",9,11:druck1 dir.Name$(0),9,19
Df    druck1 "Bitte ein wenig Geduld.....",9,27
b0    Lauf%=1:beginn%=1:Ende%=max%-1
L7    FOR modus%=0 TO 1
VS4     WHILE Lauf%=1
Jy6       Lauf%=0
ZP        FOR i% = beginn% TO Ende%
QT8         IF UCASE$(dir.Name$(i%)) > UCASE$(dir.Name$(i%+1))
             THEN
OpC             SWAP dir.Name$(i%),dir.Name$(i%+1)
5Z              SWAP dir.Typ$(i%), dir.Typ$(i%+1)
R7              Lauf%=1
SL8           END IF
DY6       NEXT i%
K8        beginn%=beginn%+1:Lauf%=0
kn        FOR i%=Ende% TO beginn% STEP -1
ad8         IF UCASE$(dir.Name$(i%)) < UCASE$(dir.Name$(i%-1))
             THEN
E5C             SWAP dir.Name$(i%),dir.Name$(i%-1)
Hn              SWAP dir.Typ$(i%), dir.Typ$(i%-1)
ZF              Lauf%=1
aT8           END IF
Lg6       NEXT i%
tA        Ende%=Ende%-1
8w4     WEND
XK2   NEXT modus%
1n0 END SUB
RW  REM Directory von Disk einlesen:
m6  '----------------------------
er  SUB LiesDisk (dir$, max%) STATIC
o92   SHARED fehler%
cf    lies% = -2
XW    dir0$ = dir$ + CHR$(0)
zC    Puffer=252
cS    Adresse = Lock(SADD(dir0$),lies%)
w1    zaehler%=0
zI    fehler%=0
Q3    IF Adresse = 0 THEN
um4     Fuelle 6,11,631,36
Ub      druck1 "Inhaltsverzeichnis existiert nicht !!",9,19
i9      FOR i%=0 TO 2000:NEXT
X9      fehler%=500
8M      EXIT SUB
wp2   END IF
2E    opt = 2^16
1X    info = AllocMem(Puffer,opt)
Hm    IF info = 0 THEN ERROR 7
D3    erfolg% = Examine%(Adresse,info)
LR    IF erfolg% = 0 THEN
5x4     Fuelle 6,11,631,36
KU      druck1 "Inhaltsverzeichnis konnte nicht untersucht werd
      en!",9,19
tK      FOR i%=0 TO 2000:NEXT
iK      fehler%=500
JX      EXIT SUB
702   END IF
1x    WHILE fehler%<>232
Gg4     dir.Name = info+8
tn      FOR schleife% = 0 TO 29
116       check% = PEEK(dir.Name+schleife%)
RA        IF check%<>0 THEN
Q48         check$ = check$+CHR$(check%)
tc6       ELSE
sY8         schleife%=29
G96       END IF
OK4     NEXT schleife%
1y      dir.Name$(zaehler%) = check$
x9      check$=""
HZ      typ = PEEKL(info+120)
Q8      IF typ < 0 THEN
yT6       dir.Typ$(zaehler%)="DATEI"
vG4     'ELSEIF zaehler% = 0 THEN
zP5      ' dir.Typ$(zaehler%) = "AKT. DIR."
4n4     ELSE
oF6       dir.Typ$(zaehler%) = "DIR"
RK4     END IF
6I      check$=""
4d      erfolg%=ExNext%(Adresse,info)
bT      IF erfolg%=0 THEN
B76       fehler%=IoErr%
c2        IF fehler% <> 232 THEN
aS8         Fuelle 6,11,631,36
Wx          druck1 "Fehler im Inhaltsverzeichnis!",9,19
Op          FOR i%=0 TO 2000:NEXT
Dp          fehler%=500
o2          EXIT SUB
cV6       END IF
I14     ELSE
```

```
nn6       zaehler%=zaehler%+1
he        IF zaehler%>max% THEN
Qp8         fehler%=232
M56       ELSE
sB8         fehler%=0
Jc6       END IF
kd4     END IF
G42   WEND
xM    CALL FreeMem(info,Puffer)
bP    CALL Unlock(Adresse)
Zs    max%=zaehler%
vx0 END SUB
vu  REM Schattenschrift f. Einleitung:
pZ  '--------------------------------
oN  SUB Schatten (text$,mode%) STATIC
AT2   POKEW WINDOW(8)+58,9
1n    textlen%=LEN(text$)
2S    x%=PEEKW(WINDOW(8)+36)
9c    y%=PEEKW(WINDOW(8)+38)
p8    COLOR 2,0
dL    CALL Move(WINDOW(8),x%+2,y%+2)
cM    CALL text(WINDOW(8),SADD(text$),textlen%)
hq    CALL SetDrMd(WINDOW(8),0)
q8    COLOR 1,0
x1    CALL Move(WINDOW(8),x%,y%)
gQ    CALL text(WINDOW(8),SADD(text$),textlen%)
Qg    IF mode%=0 THEN
V84     PRINT
6z2   END IF
t3    CALL SetDrMd(WINDOW(8),1)
EG0 END SUB
4R  REM schnelle Textausgabe im Window:
OI  '----------------------------------
w7  SUB druck1 (txt$,x%,y%) STATIC
Vs3    SHARED FensterRP
Jm     txtlen%=LEN(txt$)
09     CALL Move(FensterRP,x%,y%+7)
hi     CALL text(FensterRP,SADD(txt$),txtlen%)
MOO END SUB
Nh  REM Ausgabefelder loeschen:
4o  '--------------------------
E3  SUB Fuelle(x1%,y1%,x2%,y2%) STATIC
d01  SHARED FensterRP
ow   CALL SetAPen(FensterRP,0)
Ds   RectFill FensterRP,x1%,y1%,x2%,y2%
v4   CALL SetAPen(FensterRP,1)
UW0 END SUB
a8  REM StringInfo Struktur anlegen:
Vp  '-----------------------------
AT  SUB StrInfo(stringpuffer,buff$) STATIC
1v1  SHARED EditGroesse
El   max%= 15
Mw   IF LEN(buff$) > max% THEN buff$=LEFT$(buff$,max%)
oU   IF (max% AND 1) THEN max%=max%+1
10   EditGroesse=36+2*(max%+4)
py   stringpuffer=AllocMem(EditGroesse,65538&)
mk   IF stringpuffer=0 THEN ERROR 7
Ts   POKEL stringpuffer,stringpuffer+36
ne   POKEW stringpuffer+10,16
5R   IF buff$<>"" THEN
sI3    FOR i%=1 TO LEN(buff$)
tE5      POKE stringpuffer+35+i%,ASC(MID$(buff$,i%,1))
ns3    NEXT
fY1  END IF
mo0 END SUB
m5  REM editierten String aus Speicher lesen:
Yf  '-------------------------------------
DY  SUB LiesEdit (stringpuffer,ret$) STATIC
JD2   SHARED EditGroesse
pR    puffer.string=PEEKL(stringpuffer)
PC    ret$="":i%=0:check%=1
sS    WHILE check% > 0
GF4     check%=(PEEK(puffer.string+i%))
xT      ret$=ret$+CHR$(check%)
vx      i%=i%+1
MA2   WEND
M8    ret$=LEFT$(ret$,LEN(ret$)-1)
qR    CALL FreeMem(stringpuffer,EditGroesse)
020 END SUB
ga  REM linke Maustaste ??
dx  '--------------------
b0  SUB Maus STATIC
pg1  dummy=MOUSE(0)
Gg   WHILE MOUSE(0) = 0
VJ   WEND
790 END SUB
hS  REM Druckroutine
v2  '--------------
CQ  SUB Etikett STATIC
z1  REM # # #
Ah2   SHARED FensterAdresse,DruckFrageTxt,JaTxt,NeinTxt,Druck%
EI    Antwort=AutoRequest(FensterAdresse,DruckFrageTxt,JaTxt,Ne
      inTxt,0&,0&,300,50)
11    IF Antwort < 1 THEN
SC4     GOTO aus
A32   END IF
03    anzahl%=18
RN    esc$=CHR$(27) : init$=esc$+"@"
fz    fett$=init$+esc$+"E"+esc$+"W"+CHR$(1)
sx    abstand$=esc$+"A"+CHR$(6)
uQ    klein$=+esc$+"S0"
Zb    eng$=CHR$(15)
B8    elite$=esc$+"M"
pJ    Ende$=esc$+"A"+CHR$(13)+esc$+"2"
Bp    titel$=Druck$(0)
f2    titel2$=LEFT$(Druck$(0),13)
L1    OPEN "par:" FOR OUTPUT AS #1
2R4     PRINT#1,init$;abstand$;   :'IBM-MODE +esc$+"2";
8I      PRINT#1,klein$;eng$;
Oh      PRINT#1,SPC(29);"KEIN Schreibschutz"
Wf      PRINT#1,
sj      PRINT#1,SPC(34);"Schreibschutz"
Yh      PRINT#1,
u9      einrueck=INT((48-LEN(titel$))/2)   :REM ** Titel mittig
         **
Ga      justify=24-INT(LEN(titel$)/2+.5)
pn      PRINT#1,SPC(einrueck);titel$
11      einrueck=INT((14-LEN(titel2$))/2)   :REM ** Titel mitti
        g **
za      PRINT#1,fett$
6M      PRINT#1,SPC(einrueck);titel2$
Ab      PRINT#1,esc$+"W0";
yI      PRINT#1,init$;
fx      IF Druck% > 19 THEN
4b6       anzahl% = 32
f9        PRINT#1,abstand$;  :'IBM-MODE +esc$+"2";
sC        PRINT#1,klein$;
eX4     END IF
ut      REM ** Rahmen **
JC      PRINT#1,elite$;CHR$(32); :'IBM-MODE:chr$(201)
Er      FOR i=1 TO 15
w26       PRINT#1,CHR$(45);       :'IBM (205)
EU4     NEXT i
dv      PRINT#1,CHR$(32);         :'IBM (203)
Iv      FOR i=1 TO 15
066       PRINT#1,CHR$(45);       :'IBM (205)
w14     NEXT
M4      PRINT#1,CHR$(32)          :'IBM (187)
TH      FOR i=1 TO anzahl% STEP 2
X06       v1=INT((15-LEN(Druck$(i)))/2)
vG        n1=INT((15-LEN(Druck$(i)))/2+.5)
CQ        v2=INT((15-LEN(Druck$(i+1)))/2)
mc        n2=INT((15-LEN(Druck$(i+1)))/2+.5)
04        PRINT#1,CHR$(124);SPC(v1);Druck$(i);SPC(n1);CHR$(124
          ); :'IBM (186)
u3        PRINT#1,SPC(v2);Druck$(i+1);SPC(n2);CHR$(124)
             :'IBM (186)
Rh4     NEXT i
Y2      PRINT#1,CHR$(32);         :'IBM (200)
V8      FOR i=1 TO 15
DJ6       PRINT#1,CHR$(45);       :'IBM (205)
V14     NEXT i
oA      PRINT#1,CHR$(32);         :'IBM (202)
ZC      FOR i=1 TO 15
HN6       PRINT#1,CHR$(45);       :'IBM (205)
Zp4     NEXT i
JN      PRINT#1,CHR$(32)          :'IBM (188)
Lf      PRINT#1,klein$;
YX      PRINT#1,Ende$
QD      CLOSE #1
s3      aus:
q7      FOR i%=0 TO 32
D85      Druck$(i%)=" "
LQ4     NEXT
JL0 END SUB
(C) 1990 M&T
```

Listing. »DiskEti« (Schluß)

und läuft und läuft und läuft und

Laufschriften sind eine Methode, Informationen auszugeben. Sollten Sie vor deren Programmierung zurückschrecken, dann brauchen Sie dieses Utility. Mit »hscroll« bringen Sie Schwung in die Langeweile des CLI.

Schluß mit den Befehlen »Echo« und »Type«, es lebe die Laufschrift! Dabei haben Sie vollen Einfluß auf den scrollenden Text. Dieser kann beliebig plaziert, sowie in unterschiedlicher Breite und mit variabler Geschwindigkeit dargestellt werden.

Unser Tool »hscroll« (Listing 1) ist in Aztec-C geschrieben. Geben Sie das Listing bitte mit dem Checksummer ein. Zur Compilation tippen Sie einfach »cc +l hscroll.c« gefolgt von »ln hscroll.o -lc32« ein. Danach wird es am besten in den C-Ordner kopiert, so daß es jederzeit zur Verfügung steht. Folgende Optionen stehen zur Verfügung, die alle mit einem Leerzeichen getrennt hinter »hscroll« eingegeben werden:

Parameter für hscroll

-lx:	x ist die Position auf der x-Achse für die Laufschrift (Werte von 0 bis 639).
-ty:	y gibt die y-Position an (Werte von 1 bis 256 beim PAL-Amiga).
-wn:	setzt die Breite des Ausgabefensters auf n Pixel. Das Fenster kann maximal 640 Punkte breit sein, wenn Sie Null bei »-l« angeben.
-sn:	n legt die Geschwindigkeit fest, die Werte von 1,2,4 oder 8 annehmen kann.
-rn:	n entspricht hier der Anzahl der Wiederholungen. Bei n=1 läuft Ihr Text nur einmal durch.

Der eigentliche Text der Laufschrift wird mit einem Editor geschrieben und als ASCII-File abgespeichert. Er kann dann ebenfalls als Parameter hinter »hscroll« übergeben werden. Beispiel:

```
hscroll DateiName -l0 -t100 -w200
```

Dieser Aufruf schickt eine Laufschrift am linken Rand des Screens (etwa in der Mitte) über eine Breite von 200 Bildpunkten. Eventuelle Returns innerhalb des Ausgabe-Textes werden durch Leerzeichen ersetzt. Dies ermöglicht ungeahnte Effekte: Lassen Sie Ihre eigenen Fehlermeldungen anzeigen, beim Compilieren den Source-Code durchlaufen oder geben Sie Directories in Laufschrift aus.

Haben Sie eine der Optionen nicht verwendet, so werden Default-Werte benutzt, von denen Sie allerdings nur den für »-r« kennen müssen. Ohne »-r« gibt es eine annähernd unbegrenzte Wiederholungszahl, nämlich genau 2^{32}-1. Dies sind immerhin etwas über 4 Milliarden Wiederholungen; für die meisten Anwendungen dürfte das ausreichen. Dies heißt aber nicht, daß Sie Ihre Laufschrift jetzt eine Woche lang betrachten müssen, sie kann auch mit der Maus weggeklickt werden.

Mehr Effekt mit »Run«

Sie können »hscroll« auch problemlos als Background-Task starten, so daß die Laufschrift im Hintergrund ihrer Wege geht. Dies kann beispielsweise sinnvoll sein, wenn ein längeres Programm geladen werden soll und währenddessen eine Laufschrift das Ganze ein wenig schmücken könnte. Installieren Sie diese also zuerst mit »run hscroll« und den entspechenden Parametern (am besten in einer Batch-Datei) und erst danach kommt der eigentliche Befehl zum Laden. Dabei kann es vorkommen, daß der Ladevorgang geringfügig langsamer wird (je nach Größe der Ausgabe). Soll die Laufschrift noch kurz vor Beendigung des Ladens verschwinden, können Sie die Option »-r« verwenden.

Es gibt fünf Dinge, die Sie bei der Benutzung dieses Tools wissen sollten:

– Das Scrolling ist nicht unbedingt immer stufenlos, wenn die Laufschrift ganz oben im Bildschirm geöffnet wird. Das hängt damit zusammen, daß der Task immer auf die vertikale Austastlücke wartet. Dies ist – vereinfacht gesagt – der Moment, in dem der Elektronenstrahl sich in Zeile 0 befindet. Je weiter oben sich nun die Laufschrift befindet, desto weniger Zeit verbleibt, den Window-»RastPort« zu scrollen, ohne daß zugleich das Bild dargestellt wird. Wenn hierfür nicht genügend Zeit ist, flimmert die Textdarstellung. Der Einfachheit halber wurde auch auf das sogenannte »Double Buffering« verzichtet.

– Sind mit »run hscroll« mehrere Tasks aufgerufen worden, so kann es leicht passieren, daß Ihr CLI Task sozusagen »out of CPU« gerät, also keine Rechenzeit mehr bekommt. Das liegt daran, daß die Laufschrift-Tasks bei Aufruf die höchstmögliche Priorität bekommen, um ein Ruckeln zu verhindern. Man mache sich klar, daß die vertikale Austastlücke etwa 50 mal pro Sekunde auftritt und »hscroll« ebensooft den Prozessor benötigt.

– Es können nur 8 x 8-Punkt Zeichensätze verwendet werden. Wer also in seinem CLI andere benutzen möchte, der sollte erst einmal diese Einschränkung überprüfen. Aus diesem Grunde beträgt die Höhe des geöffneten Scroll-Fensters auch immer 8 Pixel.

– Laufschriften sind nichts anderes als kleine Fenster. Sie können von anderen Fenstern überlappt werden und dann einfach verschwinden.

– Jedes Programm, das längere Blitter-Operationen veranlaßt, ist ein potentieller Störfaktor bezüglich des Filescrollings. Schließlich benutzt auch »hscroll« den Blitter zum Scrolling. Irgendwann aber darf jeder Task mit auch noch so kleiner Priorität mal den Blitter verwenden, was ein leichtes Rukkeln zur Folge hat. Dies kann nicht umgangen werden, wenn man sich konsequent an die Multitasking-Konventionen halten will.

Zur Beschleunigung der Textausgabe gibt es auf dem Amiga einige nützliche Utilities, unter anderem auch »Blitz-Fonts!« und »FastFonts«, das sich auf der neuen Workbench 1.3 befindet. Diese wirken sich auch günstig auf unsere laufenden Texte aus. Sie sind also unbedingt zu empfehlen, vor allem wegen der geringeren Rechenzeit.

Um dem Laufschrift-Task ein möglichst ruckelfreies Dasein zu bescheren, wird mittels »SetTaskPri(FindTask(0),127);« die Priorität desselben auf den größtmöglichen Wert gesetzt. »FindTask(0)« durchsucht die Liste der gerade ablaufenden Tasks und übergibt den entsprechenden Zeiger auf die TaskStruktur. Nach Beendigung der Hauptschleife wird die Priorität dann auf einen normalen Wert zurückgesetzt. Es werden die üblichen Libraries geöffnet, die für die Verwaltung von Fenstern und Grafik notwendig sind. Um den einzulesenen ScrollText dynamisch verwalten zu können, muß dessen Länge bekannt sein.

Besser geht's mit BlitzFonts!

Es wird daher Speicher für die Daten des »FileInfoBlocks« reserviert, aus dem man nach Untersuchung des Files die Länge desselben entnehmen kann (»FileInfoBlock->fib_Size«).

und läuft und läuft und läuft und

Sozusagen als Unterlage für die Laufschrift wird ein einfaches Fenster »ohne alles« geöffnet, welches im übrigen genau ein Gadget zum Wegklicken enthält.

Die Hauptschleife wartet zunächst mit »WaitTOF();« auf die nächste Austastlücke und scrollt dann den gesamten »RastPort« des Fensters um p Pixels (je nach eingestellter Geschwindigkeit) nach links. Mit Hilfe der Text-Funktion wird dann der neu hinzugekommene Teil neu geschrieben. Das ist in diesem Fall äußerst einfach, weil Operationen im RastPort grundsätzlich nicht über den Rand hinaus gehen, also nichts außerhalb unseres Fensters verändern. Eventuelle Reste werden einfach abgeschnitten.

Smooth mit WaitTOF()

Die Hauptschleife wird solange durchlaufen, bis entweder die Variable für die Wiederholungen abgelaufen ist oder eine Nachricht am Port des Fensters anliegt. Auch dieser Fall ist leicht zu handhaben, da wir nur ein Gadget zur Verfügung haben, das den IDCMP des Fensters ansprechen kann; damit können wir uns eine Auswertung der Nachricht sparen.

Möglicherweise stört Sie die Beschränkung von »hscroll« auf eine Zeile? Die Lösung heißt »vscroll« (Listing 2). Dieses Programm ist in den wesentlichen Punkten genauso aufgebaut. Das Textfile wird jetzt allerdings nach der längsten enthaltenen Zeile abgesucht. Diese bestimmt die Breite des Ausgabefensters. Falls der rechte Rand des Screens überschritten wird, bricht das Programm vorher ab. Selbstverständlich arbeitet »vscroll« auch mit exotischen Formaten wie der 704 Pixel breiten Workbench oder dem PD-Programm »VScreen-Handler« zusammen.

Bitte beachten Sie den Listingausschnitt von Zeile 163 bis Zeile 166. Hier wird dem Betriebssystem suggeriert, es würde nur eine Bitplane bestehen. Durch diesen kleinen Trick erreichen wir eine geringere Belastung des Blitters für die Scroll-Operationen. Vor Beenden des Programms werden die Bitplanes wieder auf Normalwerte gesetzt.

»vscroll« können Sie mit dem Aztec (»cc vscroll.c« gefolgt von »ln vscroll.o -lc«) oder mit dem Lattice (»lc -w vscroll«) compilieren.

Vielleicht fallen Ihnen selbst noch einige Verbesserungen ein, wie zum Beispiel verschiedene Schriftarten, per Tastendruck verschiebbares Fenster oder ähnliches? Wir wünschen Ihnen auf jeden Fall viel Spaß beim Experimentieren mit diesem Utility. *Christian Wolf/kn*

Programmname:	hscroll
Computer:	A500, A1000, A2000 mit Kickstart 1.2, 1.3
Sprache:	Aztek-C
Compiler:	Version 3.6
Aufrufe:	cc +l hscroll.c ln hscroll.o -lc32

```
Programmautor: Christian Wolf
-----------------------------

#include exec/types.h
#include intuition/intuition.h
#include intuition/intuitionbase.h
#include exec/exec.h
#include exec/execbase.h
#include libraries/dos.h
#include libraries/dosextens.h
#include graphics/gfxmacros.h
#include functions.h
#define NOBORDER NULL
#define NOTEXT NULL
/* Gadget zum Abklicken... */
struct Gadget ClickGadget = { NULL,0,0,0,8,GADGHNONE,RELVER
IFY,
                                BOOLGADGET,NOBORDER,0,NOTEXT,
                                0,0,NULL,0};
/* Einfaches Fenster ohne alles... */
struct NewWindow NewScrollWindow = {
        0,230,640,8,0,1,
        GADGETUP,
        BORDERLESS,
        &ClickGadget,NULL,
        NULL,
        NULL,NULL,
        0,0,0,0,
        WBENCHSCREEN };
/* Das Übliche... */
struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
/* Default scrolltext */
char *DefaultText= "Hscroll 1.0 (w) 1989 by Christian Wolf.
 Options: -lx and -ty"
                      " puts your scroll to equivalent x and y
                       position. Use -w to"
                      " select the window's width. -s is used
                      to have different"
                      " scroll speeds. Values allowed are: 1,2
                      ,4,8. Add a filename"
                      " to your parameter list to have your ow
                      n text. -rn lets"
                      " the text repeat n times. To remove the
                       scrolltext, just"
                      " click it. Have much fun...
                                       ";
main(argc,argv)
 int argc;
 char **argv;
{
 struct Window *ScrollWindow;
 struct RastPort *ScrollRastPort;
 short l=160,t=248,w=320,s=2,p=0,c=0,sl=strlen(DefaultText)
 ;
 long r=0xffffffff,Size;
 struct FileHandle *ThisFile;
 struct FileLock *MyLock;
 char *ScrollText=DefaultText,*MyPtr;
 struct FileInfoBlock *FileInfoBlock;
 /* Damit es nicht so ruckelt -> höchste Taskprorität */
 SetTaskPri(FindTask(0),127);
 /* FileInfoBlock öffen... */
 FileInfoBlock=AllocMem(sizeof(struct FileInfoBlock),MEMF_P
 UBLIC|MEMF_CLEAR);
 if (!FileInfoBlock) exit(0);
 /* parameter liste durchgehen */
 argc--;
 argv++;
 while (argc--)
        {
         if (**argv=='-')
             {
              (*argv)++;
              switch((*(*argv)++))
                    {
                     case 'w':
                     w=atoi(*argv);
                     break;
                     case 'l':
                     l=atoi(*argv);
                     break;
                     case 't':
                     t=atoi(*argv);
                     break;
                     case 's':
                     s=atoi(*argv);
                     break;
                     case 'r':
                     r=atoi(*argv);
                     break;
                    }
             }
```

Listing 1. »hscroll.c« zaubert Ihnen eine horizontale Laufschrift auf den Bildschirm

```c
        else
           {
              /* TextFile laden... */
              if (ThisFile=Open(*argv,MODE_OLDFILE))
                 {
                   MyLock=Lock(*argv,ACCESS_READ);
                   Examine(MyLock,FileInfoBlock);
                   Size=FileInfoBlock->fib_Size;
                   if (MyPtr=ScrollText=AllocMem(Size,MEMF_PUBLIC))
                        {
                          Read(ThisFile,ScrollText,Size);
                          MyPtr[Size]=0;
                          /* eventuelle returns entfernen... */
                          while (*MyPtr)
                                 {
                                  if (*MyPtr == 0x0a) *MyPtr++=' ';
                                  else MyPtr++;
                                 }
                          UnLock(MyLock);
                          Close(ThisFile);
                        }
                   else ScrollText=DefaultText;
                 }
              sl=strlen(ScrollText);
           }
        argv++;
      }
/* Libries öffnen... */
IntuitionBase=(struct IntuitionBase *)OpenLibrary("intuition.library",0);
GfxBase=(struct GfxBase *)OpenLibrary("graphics.library",0);
/* Werte für Fenster und Gadget initialisieren */
NewScrollWindow.LeftEdge=l;
NewScrollWindow.TopEdge=t;
NewScrollWindow.Width=w;
ClickGadget.LeftEdge=0;
ClickGadget.TopEdge=0;
ClickGadget.Width=w;
/* Fenster öffnen */
if (!(ScrollWindow=OpenWindow(&NewScrollWindow))) exit(0);
/* Rasterport holen */
ScrollRastPort=ScrollWindow->RPort;
SetAPen(ScrollRastPort,3);
SetBPen(ScrollRastPort,0);
SetDrMd(ScrollRastPort,JAM1);
/* Hauptschleife -> Scrollen, solange r!=0 oder UserPort leer */
while (!GetMsg(ScrollWindow->UserPort) && r)
 {
  /* Auf den VBL warten */
  WaitTOF();
  /* Grafik um s Pixels verschieben... */
  ScrollRaster(ScrollRastPort,s,0,0,0,w-1,7);
  Move(ScrollRastPort,w-s-p,6);
  if (sl <= c)
      {
       c=0;
       r--;
      }
  /* Nachfolgenden Text reinschreiben */
  Text(ScrollRastPort,&ScrollText[c],1);
  p+=s;
  if (p > 8-s)
      {
       p=0;
       c++;
      }
 }
/* Alles schließen... */
CloseWindow(ScrollWindow);
CloseLibrary(IntuitionBase);
CloseLibrary(GfxBase);
if (ScrollText!=DefaultText) FreeMem(ScrollText,sl);
if (FileInfoBlock) FreeMem(FileInfoBlock,sizeof(struct FileInfoBlock));
SetTaskPri(FindTask(0),0);
}
```


Listing 1. (Schluß)

Programmname:	vscroll
Computer:	A500, A1000, A2000 mit Kickstart 1.2, 1.3
Sprache:	C
Compiler:	Aztec oder Lattice
Aufrufe:	siehe Text

Programmautor: Christian Wolf

```c
#include <intuition/intuitionbase.h>
#include <libraries/dos.h>
#include <exec/memory.h>
#include <graphics/gfxbase.h>
#ifdef LATTICE
#include <proto/all.h>
#else
#include <functions.h>
#endif
#define fib FileInfoBlock
#define fibsize (long)sizeof(struct fib)
#define NOBORDER NULL
#define NOTEXT NULL
/* Gadget zum Abklicken... */
struct Gadget ClickGadget = { NULL,0,0,0,0,GADGHNONE,RELVERIFY,
                              BOOLGADGET,NOBORDER,0,NOTEXT,
                              0,0,NULL,0};
/* Einfaches Fenster ohne alles... */
struct NewWindow NewScrollWindow = {
        0,230,640,8,0,1,
        GADGETUP|GADGETUP,
        BORDERLESS,
        &ClickGadget,NULL,
        NULL,
        NULL,NULL,
        640,200,640,200,
        WBENCHSCREEN };
/* Das Übliche... */
struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
struct Table { char *Line;
               long Len; };
struct Table DefaultText[] = { "              VScroll 1.0         ",37,
                              "  (c) 1989 by Markt und Technik     ",37,
                              "    written by Christian Wolf       ",37,
                              "     currently resident at          ",37,
                              "       Spandauer Str. 104L          ",37,
                              "         1 Berlin 20                ",37,
                              "       Tel.: 030/3660450            ",37,
                              "  Usage: vscroll textfile [options] ",37,
                              "  Use -w to modify window's width   ",37,
                              "  Use -s number of jiffies 1,2,4,8  ",37,
                              "  Use -l and -t to position window  ",37,
                              "  Use -r for no. of repetitions     ",37,
                              "  Use -h for a new height of window ",37,
                              "                                    ",37,
                              "                                    ",37,
                              "                                    ",37,
                              "                                    ",37,0 };
struct BitMap ScrollBitMap;
void main(argc,argv)
 char **argv;
 int argc;
{
```

```
struct Table *ScrollText=DefaultText,*cl=DefaultText;
char *MyPtr,*LoadText=0;
struct Window *ScrollWindow;
struct RastPort *ScrollRastPort;
long l=160,t=60,w=37*8,s=1,p=0,c=0,h=100,sl;
long r=0xffffffff,Size,Lines;
struct fib *fib;
struct FileHandle *ThisFile;
struct FileLock *MyLock;
int i;
/* Damit es nicht so ruckelt -> höchste Taskprorität */
SetTaskPri(FindTask(0L),127L);
/* parameter liste durchgehen */
argc--;
argv++;
while (argc--)
      {
       if (**argv=='-')
          {
           (*argv)++;
           switch((*(*argv)++))
                {
                 case 'h':
                 h=atoi(*argv);
                 break;
                 case 'l':
                 l=atoi(*argv);
                 break;
                 case 't':
                 t=atoi(*argv);
                 break;
                 case 's':
                 s=atoi(*argv);
                 break;
                 case 'r':
                 r=atoi(*argv);
                 break;
                }
          }
       else
          {
           if (ScrollText==DefaultText)
              {
               /* TextFile laden... */
               if (ThisFile=(struct FileHandle *)Open(*argv,MODE_OLDFILE))
                  {
                   /* fib öffen... */
                   if (fib=(struct fib *)AllocMem(fibsize,MEMF_PUBLIC))
                      {
                       MyLock=(struct FileLock *)Lock(*argv,ACCESS_READ);
                       Examine(MyLock,fib);
                       UnLock(MyLock);
                       Size=fib->fib_Size;
                       if (LoadText=(char *)AllocMem(Size,MEMF_PUBLIC))
                          {
                           Read(ThisFile,LoadText,Size);
                           LoadText[Size-1]=0;
                           for (MyPtr=LoadText,Lines=2;*MyPtr;MyPtr++)
                               {
                                if (*MyPtr == 0x0a) Lines++;
                               }
                           if (!(cl=ScrollText=(struct Table *)AllocMem(Lines<<3,MEMF_PUBLIC|MEMF_CLEAR))) exit(0);
                           for (MyPtr=LoadText,cl++->Line=LoadText;*MyPtr;)
                               {
                                if (*MyPtr == 0x0a)
                                   {
                                    *MyPtr=0;
                                    cl++->Line=++MyPtr;
                                   }
                                else MyPtr++;
                               }
                           cl->Line=0;
                           for (cl=ScrollText,w=0;cl->Line;cl++)
                               {
                                if ((cl->Len=strlen(cl->Line)) > w) w=cl->Len;
                               }
                           w<<=3;
                           cl=ScrollText;
                          }
                       else ScrollText=DefaultText;
                       FreeMem(fib,fibsize);
                      }
                   else ScrollText=DefaultText;
                   Close(ThisFile);
                  }
              }
          }
       argv++;
      }
/* Libries öffnen... */
IntuitionBase=(struct IntuitionBase *)OpenLibrary("intuition.library",0L);
GfxBase=(struct GfxBase *)OpenLibrary("graphics.library",0L);
/* Werte für Fenster und Gadget initialisieren */
NewScrollWindow.LeftEdge=l;
NewScrollWindow.TopEdge=t;
NewScrollWindow.Width=w;
NewScrollWindow.Height=h;
ClickGadget.LeftEdge=0;
ClickGadget.TopEdge=0;
ClickGadget.Height=h;
ClickGadget.Width=w;
/* Fenster öffnen */
if (ScrollWindow=(struct Window *)OpenWindow(&NewScrollWindow))
   {
    /* Rasterport holen */
    ScrollRastPort=ScrollWindow->RPort;
    SetAPen(ScrollRastPort,1L);
    SetBPen(ScrollRastPort,0L);
    SetDrMd(ScrollRastPort,JAM1);
    /* BitMap kopieren */
    ScrollBitMap=*ScrollRastPort->BitMap;
    ScrollRastPort->BitMap=&ScrollBitMap;
    ScrollBitMap.Depth=1;
    h+=6;
    while (!GetMsg(ScrollWindow->UserPort) && r)
         {
          WaitTOF();
          ScrollRaster(ScrollRastPort,0L,s,0L,0L,w-1,h-1);
          Move(ScrollRastPort,0L,h-s-p);
          Text(ScrollRastPort,cl->Line,cl->Len);
          p+=s;
          if (p > 8-s)
             {
              p=0;
              cl++;
              if (!cl->Line)
                 {
                  cl=ScrollText;
                  r--;
                 }
             }
         }
    ScrollBitMap.Depth=2;
    /* alles schließen */
    CloseWindow(ScrollWindow);
   }
CloseLibrary((struct Library *)IntuitionBase);
CloseLibrary((struct Library *)GfxBase);
if (ScrollText!=DefaultText) FreeMem(ScrollText,Lines<<3);
if (LoadText) FreeMem(LoadText,Size);
SetTaskPri(FindTask(0L),0L);
}
```


Listing 2. Mit »vscroll« bewegen Sie ein ganzes Textfile vertikal über den Screen

Eine uni Dateiver

Eine Dateiverwaltung in Basic? »Langsam und unkomfortabel«? Weit gefehlt: Die »Uni-Datei« ist schnell und besticht mit einer Benutzerführung, deren Komfort so manchen C-Programmierer vor Neid erblassen läßt.

Von Rainer Karwelies und Andreas Lietz

Kennen Sie den »Basic-Effekt«? Da benutzt man ein selbstgeschriebenes oder abgetipptes Programm in Amiga-Basic und erntet von den C-, Assembler- und Modula-Freunden nur ein Lächeln, wenn man ihnen das Programm vorführt. Mit der »Uni-Datei« wird das anders. Dieses Programm benutzt die Fähigkeiten des Amiga zur komfortablen Benutzerführung so ausgiebig, daß man fast vergessen könnte, in welcher Sprache es geschrieben ist! Der große Vorteil dieses Programmes ist, daß Sie das Programm jederzeit an Ihre Bedürfnisse anpassen können, wenn Sie über Basic-Kenntnisse verfügen.

Die Uni-Datei ist eine sequentielle Dateiverwaltung. Sie kann Datensätze mit maximal 18 Datenfeldern verwalten; die Anzahl der Datensätze hängt vom verfügbaren Speicherplatz und von der Anzahl der Felder ab. Ist die Feldanzahl höher, wird die Anzahl der Datensätze geringer und umgekehrt.

Das Programm ist darauf ausgelegt, dem Benutzer einen möglichst großen Komfort zu bieten. Mit Hilfe der Maus und kassettenrecorder-ähnlichen Tasten (Bild 1 unten) wählen Sie einzelne Datensätze schnell an. Es werden viele Menüs verwendet, mit deren Hilfe alle Funktionen des Programmes aufgerufen werden. In einer Datei kann nach einem bestimmten Feld gesucht und sortiert werden; die Daten lassen sich automatisch auf Etiketten oder als Liste drucken. Es ist jederzeit möglich, die Feldnamen zu ändern. Mit einem Hilfsprogramm ändern Sie sogar die Anzahl der Felder oder deren Reihenfolge, ohne daß dabei Daten verlorengehen.

Der ganze Komfort hat nur einen Preis: Das Programm ist zu lang zum Abtippen. Deshalb wird es auf der Programmservice-Diskette zu diesem Heft veröffentlicht. Auf dieser Diskette finden Sie die beiden Programmversionen »Uni__0.5MB« und »Uni__1MB« sowie die Ladeprogramme »Lader-Uni__0.5 MB« und »Lader-Uni__1MB«. Die erste Version ist für Besitzer eines Amiga mit 512 KByte Speicher gedacht; wer mehr Speicher hat, verwendet die zweite Version. Außerdem ist auf der Diskette das Hilfsprogramm »UniUtil«, mit dem Sie bei vorhandenen Dateien die Feldanzahl ändern und die Reihenfolge der Felder vertauschen können.

Um mit der Uni-Datei zu arbeiten, sollten Sie sich erst einmal eine Arbeitsdiskette formatieren. Öffnen Sie dann die Schublade »Uni-Datei« auf der Programmservice-Diskette. Haben Sie mehr als 512 KByte Speicher, so kopieren Sie nun alles außer »Lader-Uni__0.5MB« und »Uni__0.5MB« auf Ihre Diskette. Wenn Sie mit 512 KByte Speicher arbeiten, kopieren Sie alles außer »Lader-Uni__1MB«, »Uni-Datei 1MB«, »exec.bmap« und »dos.bmap« auf Ihre Arbeitsdiskette. Für alle Versionen muß schließlich noch AmigaBasic auf die Diskette kopiert werden – Sie finden es auf Ihrer Extras-Diskette. Nun haben Sie eine Arbeitsdiskette, auf der viel Platz für Daten zur Verfügung steht. Bevor Sie mit der Uni-Datei arbeiten, muß sie noch für Ihre Wünsche angepaßt werden. Dazu laden Sie das Programm mit einem Doppelklick auf »Lader-Uni__1MB« bzw. »Lader-Uni__0.5MB«. Sobald die blinkende Abfragebox (Bild 2) erscheint, drücken Sie <CTRL c> und tippen »LIST«. Nun wird der Anfang des Programmes auf dem Bildschirm angezeigt. Am einfachsten geht die Anpassung, wenn Sie das Fenster, in dem das Programm angezeigt wird, auf die volle Bildschirmgröße bringen.

Sie sehen jetzt in der dritten Zeile des Listings zwei »CLEAR«-Befehle. Sie dienen dazu, für das Programm und die Daten ausreichend Speicher zu reservieren. Für 512-KByte- und 1-MByte-Amigas sind sie in der entsprechenden Version bereits richtig eingestellt. Sollten Sie später an die Grenzen des Speichers stoßen, können Sie beim zweiten CLEAR eine höhere Zahl einsetzen – wieviel höher sie sein kann, hängt davon ab, welche Programme sonst noch in Ihrem Rechner Speicher belegen. Haben Sie mehr als 1 MByte Speicher zur Verfügung, können Sie den Wert natürlich kräftig hochsetzen.

Einige Zeilen weiter unten sehen Sie die Zuweisung »Max=250«. Diese Zahl ist die maximale Zahl der Datensätze und für normale Anwendungen völlig ausreichend. Haben Sie aber viel Speicher oder Datensätze mit sehr wenigen Feldern, so können Sie diese Zahl hochsetzen – meldet AmigaBasic daraufhin »Out of Memory«, so müssen Sie »Max« wieder reduzieren. Umgekehrt sollten Sie »Max« niedriger festlegen, wenn Ihre Datensätze 18 Felder haben und Sie mit 512 KByte Speicher auskommen müssen.

Nun folgt die Zuweisung

```
DName$ = "Adressen"
```

Hier setzen Sie statt »Adressen« den Namen der Datei ein, die Sie am häufigsten nutzen. Dieser Name wird beim Laden von Dateien vom Programm vorgegeben (kann aber natürlich jederzeit überschrieben werden). Schließlich folgt die Anweisung

```
Pfad$ = "df1:Daten"
```

```
Dateiname:Df1:Daten/Hardcopy System: 2637000 Basic: 39131

Datensatz Nr. 2 von 2                 Datum 12.08.1988  17:47:46

Herr/Frau    : Herrn
Vorname      : Peter
Name         : Pan
Strasse      : Rübezahlstr. 127
PLZ          : 8000
Ort          : München 12
Bezirk       : Südbayern
Vorwahl      : 089
Telefon      : 1234567
Bemerkung 1  : Guter Kunde; tendiert zu unfreundlichem Auftreten
Bemerkung 2  : Gute Beziehungen zu Schweden
Kundennummer : 1254653-236
Auftragsnr.  : 12-45/78
Let. Auftrag : 12.6.89
Kredit       : 3000,-
Rabatt       : 0 %
Skonto       : 3 %
Satz OK      : Ja

 |<   <<<   <<   <   Ja   Nein   >   >>   >>>   >|
```

Bild 1. Die Uni-Datei in Aktion. Den Aufbau Ihrer Datei bestimmen Sie selbst – hier ein Beispiel.

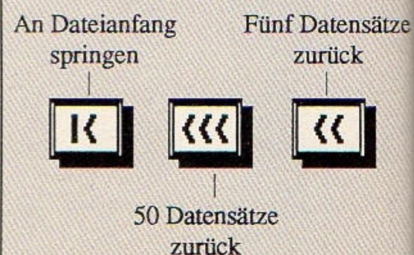


Bild 3. So verwenden Sie die

verselle waltung

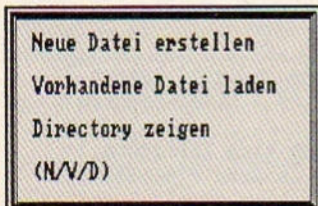


Bild 2. Diese blinkende Box erscheint beim Start der Uni-Datei

Hier wird angegeben, wo die Daten vom Programm abgelegt werden. Bitte stellen Sie diese Variable entsprechend Ihren Wünschen ein. Speichern Sie jetzt das Programm und starten Sie es erneut.

Dazu klicken Sie das jeweilige Lader-Programm an und warten, bis der Bildschirm wie in Bild 2 aussieht. Da Sie bisher noch keine Datei erzeugt haben, tippen Sie »n«, um eine neue Datei zu erstellen. Nun müssen Sie die Anzahl der Datenfelder und die Bezeichnung der einzelnen Felder sowie den Namen der Datei eingeben. Ist das erledigt, erscheint der Hauptbildschirm von Uni-Datei (Bild 3). Die von Ihnen eingegebenen Feldnamen werden links oben angezeigt. In der Kopfzeile des Bildschirms sehen Sie den vorher gewählten Pfadnamen, den Dateinamen, den freien Systemspeicher, die Uhrzeit und das Datum. Darunter stehen die Nummer des angezeigten Datensatzes und die Anzahl der Datensätze, die im Speicher stehen. Am unteren Bildschirmrand ist eine Reihe von »Schaltern«, die sich mit der Maus anklicken lassen. Die Schalter mit den Pfeilen werden verwendet, um den angezeigten Datensatz zu wechseln – ähnlich wie die Tasten bei einem Kassettenrecorder. In Bild 3 wird gezeigt, wozu Sie welchen Schalter verwenden. Die beiden mittleren Schalter, »Ja« und »Nein«, werden vom Programm bei Abfragen benutzt.

Alle Funktionen der Uni-Datei werden über Menüs mit der Maus ausgewählt. Um Datensätze in Ihre Datei einzugeben, wählen Sie aus dem ersten Menü, »Datei«, den Menüpunkt »Datei pflegen«. Das Programm fragt nun nach den neuen Inhalten für Ihre Datei. Jedes Datenfeld darf höchstens 60 Zeichen lang sein – zu lange Eingaben werden automatisch auf diese Länge gekürzt. Mit Hilfe der Funktionstastenbelegung (wird später erklärt) können Sie sich die Eingabe von oft wiederkehrenden Daten vereinfachen. Die Eingabe wird abgebrochen, indem Sie im ersten Feld eines Datensatzes zuerst <F1> und dann <RETURN> drücken.

Bevor Sie mit Ihrer Datei arbeiten, sollten Sie sie erst einmal speichern. Das geschieht mit einem der »Speichern«-Befehle aus dem Datei-Menü.

Frühzeitig Daten sichern

Wählen Sie »Speichern«, so wird die Datei unter dem Namen gespeichert, den Sie beim Erzeugen der Datei angegeben hatten. »Speichern als« erlaubt Ihnen, einen neuen Namen für die Datei einzugeben. »Speichern privat« speichert die Datei mit einem Paßwort – aber Vorsicht! Der Paßwortschutz ist so einfach, daß Sie dieses Schutzverfahren auf keinen Fall für wirklich geheime Daten verwenden sollten.

Kurz noch zu den weiteren Befehlen des Datei-Menüs: »Datei laden« lädt eine neue Datei. Das Programm erkennt automatisch, ob Sie ihre bisher verwendete Datei gesichert haben und macht eine Sicherheitsabfrage, wenn das nicht so ist. Mit »Datei anhängen« hängen Sie eine Datei an die Datei im Speicher an. Dabei muß die Anzahl der Felder natürlich bei beiden Dateien übereinstimmen. Außerdem darf die maximale Datensatzzahl nicht überschritten werden.

Um eine Datei in einem anderen Verzeichnis auf der Diskette zu verwenden, wählen Sie »Disk-Pfad ändern« und geben den neuen Pfadnamen ein. <RETURN> übernimmt den alten, falls Sie doch nichts ändern möchten. Einen neuen Dateinamen geben Sie mit »Dateinamen ändern« ein, und »Directory zeigen« zeigt – wer hätte das gedacht? – das Inhaltsverzeichnis des gewählten Pfades.

Sortieren à la carte

Schließlich können Sie mit »Neustart« das Programm von vorne beginnen (beispielsweise wenn Sie sich total verfranzt haben oder um mit einer anderen Datei zu arbeiten) und mit »Ende« das Programm beenden – in beiden Fällen werden alle Daten gelöscht.

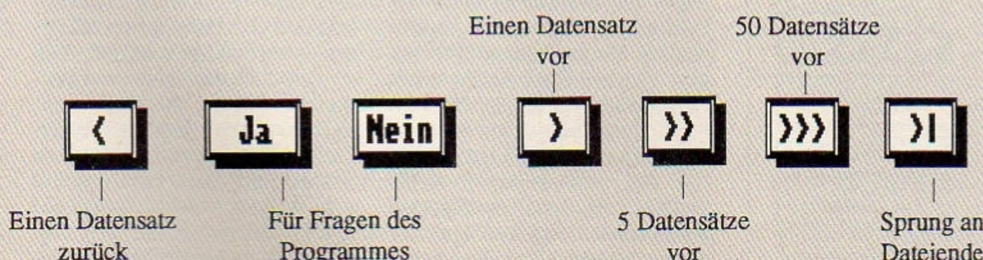
Zurück zu Ihrer Datei. Das »Datensatz«-Menü hilft Ihnen bei der Verwaltung Ihrer Daten. Hier können Sie einen Datensatz löschen, duplizieren und editieren. Diese Befehle beziehen sich immer auf den Datensatz, der gerade auf dem Bildschirm steht. Beim Löschen wird der letzte Datensatz der Datei an die Stelle des gelöschten Datensatzes geschrieben. »Duplizieren« kopiert den aktuellen Datensatz an das Ende der Datei. Um den aktuellen Datensatz zu ändern, wählen Sie »editieren« und geben die neuen Feldinhalte ein – soll ein Feld nicht verändert werden, drücken Sie einfach <RETURN>.

»Gehe zu Datensatz Nummer« bringt den Datensatz mit der eingegebenen Nummer auf den Bildschirm. Um die Feldnamen zu ändern, wählen Sie »Feldnamen editieren« – das Ändern funktioniert ähnlich wie bei »Datensatz editieren«.

Sehr wichtig für eine Dateiverwaltung sind die Sortierfunktionen. Die Uni-Datei kann Ihre Daten nach jedem Feld aufsteigend und absteigend sortieren. Ein Beispiel: Sie wollen Ihre Adreßdatei alphabetisch nach dem Nachnamen sortieren. Dazu wählen Sie »Aufsteigend sortieren«. Es erscheint ein kurzer Hinweis, in dem Sie gefragt werden, ob Sie wirklich sortieren wollen. Klikken Sie auf »Ja«. Jetzt müssen Sie dem Programm sagen, nach welchem Feld es sortieren soll: Dazu wählen Sie aus dem »Suche nach«-Menü den Namen des Feldes aus, nach dem sortiert werden soll. Während des Sortiervorgangs wird grafisch angezeigt, wie groß der schon sortierte Anteil der Datei ist. Das absteigende Sortieren funktioniert nach dem gleichen Muster.

Nachdem Sie jetzt schon einmal in das »Suche nach«-Menü hineingeschnuppert haben, wissen Sie bereits, daß in diesem Menü alle Feldnamen Ihrer Datei stehen. Wenn dieses Menü nicht gerade zum Sortieren »mißbraucht« wird, dient es zum Suchen von Daten. Dazu wählen Sie einfach den Namen des Feldes, nach dem Sie suchen wollen. Die Uni-Datei fragt nun, nach was Sie suchen wollen; geben Sie den Ausdruck an, nach dem Sie suchen. Um hier abzubrechen, tippen Sie wie üblich <F1> und <RETURN>. Beim Suchen gilt folgende Regel: Wenn der Suchbegriff mit dem Anfang des Datenfeldes übereinstimmt, wird der Datensatz angezeigt. Um weiterzusuchen, nachdem ein Datensatz gefunden wurde, wählen Sie »Weitersuchen«.

Was nutzt die beste Dateiverwaltung, wenn man die Daten nicht ausdrucken kann? Nichts. Deshalb sollten Sie sich jetzt einmal das Drucker-Menü der Uni-Datei ansehen. Hier haben Sie zunächst einmal die Möglichkeit, den Datensatz auszudrucken, der gerade auf dem Bildschirm steht. Wählen Sie »Selektierte Datensätze«, so müssen Sie wie beim Suchen einen Feldnamen auswählen und einen Suchbegriff eingeben. Das Programm druckt dann alle Datensätze, die mit diesem Suchkriterium übereinstimmen. Um alle Datensätze



»Knöpfe« am unteren Bildschirmrand. Durch Anklicken wird ihre Funktion aktiviert.

zu drucken, verwenden Sie den Befehl »Alle Datensätze«.

Die Drucker-Parameter lassen sich bei diesem Programm sehr komfortabel einstellen. Dazu wählen Sie »editieren« aus dem »Drucken«-Menü. Es erscheint ein großes Fenster (Bild 4), in dem Sie alle Drukkerparameter einstellen. Das geschieht, indem Sie mit der Maus auf die Schalter rechts im Fenster klicken. Die meisten Dinge in diesem Fenster erklären sich von selbst. Bei »Etikettenhöhe« stellen Sie ein, wie viele Felder der Datei untereinander gedruckt werden sollen – stellen Sie z.B. die Etikettenhöhe auf drei, werden nur die ersten drei Felder jedes Datensatzes gedruckt. Setzen Sie den Schalter bei »Informations-Zeile drucken« auf »ja«, wird an den

Dateien wie gedruckt

Anfang des Ausdrucks eine Titelzeitele mit Dateinamen und Datum gesetzt. Interessant ist die Frage »Datenfelder nebeneinander in eine Zeile drukken«. Damit machen Sie Tabellen aus Ihren Daten. Wenn Sie den Schalter dazu auf »Ja« schalten, erscheint ein neues Fenster (Bild 5). Hier stellen Sie ein, wie breit jedes Datenfeld in der Tabelle gedruckt werden soll. Dazu bewegen Sie den blinkenden Stern mit Hilfe der beiden Pfeile zu dem Feldnamen, dessen Breite in der Tabelle Sie ändern wollen. Mit den Rechts- und Linkspfeilen ändern Sie die Breite – der »+/-«-Schalter macht größere Änderungen. Um ein Feld nicht zu drucken, setzen Sie den Wert auf Null. Alle Werte werden vom Programm addiert und mit den Abständen zwischen den Tabellenspalten verrechnet. Das Ergebnis dieser Rechnung wird bei »Sie haben eine Zeilenbreite von x eingestellt« angezeigt. Achten Sie darauf, daß Ihr Drucker so breite Zeilen in der eingestellten Schrift drucken kann, sonst gibt es Probleme beim Drucken. Mit einem Klick auf »Ok« gelangen Sie zurück zur Druckerparameter-Einstellung; ein weiterer Klick auf »Ok« in diesem Fenster verläßt die Einstellung der Druckerparameter.

Im »Service«-Menü gibt es etwas sehr praktisches: die Funktionstastenbelegung. Dazu wählen Sie »F2 bis F10 belegen«. Hier werden diese Tasten mit einem beliebigen Text mit bis zu 60 Zeichen Länge belegt. Die F-Tasten-Belegung ist im Menüpunkt »Datei pflegen« und »Datensatz editieren« benutzbar. Die Anweisung »Status« im Service-Menü zeigt die Speicherbelegung des Programmes, und mit »Version« erhalten Sie eine Information über den Autor, Rainer Karwelies.

Sollten Sie einmal die Anzahl der Felder Ihrer Datei oder deren Reihenfolge verändern wollen, laden Sie das Programm »UniUtil«. Bevor Sie es anwenden können, müssen Sie eventuell den Pfadnamen am Anfang des Programmes ändern. Ist das Programm korrekt gestartet worden, erscheint der Hinweis »Bitte mit der Maus einen Menüpunkt anwählen«. Um die Anzahl der Felder einer Datei zu erhöhen, wählen Sie den entsprechenden Menüpunkt. Daraufhin wird nach dem Dateinamen gefragt. Nachdem Sie ihn eingegeben haben, erscheint die Frage »Wie viele Felder soll ich an die x vorhandenen anhängen (Max = x)«. Geben Sie ein, wie viele Felder Sie zusätzlich benötigen. Dabei können Sie maximal die Anzahl angeben, die vorher bei »Max« angezeigt wurde. Nun fragt das Programm nach dem Namen des oder der zusätzlichen Felder und erweitert die Datei entsprechend. Zum Speichern der Datei wählen Sie »Geänderte Datei sichern als«.

Das Löschen einzelner Felder funktioniert ähnlich: Sie wählen »Anzahl Datenfelder verringern« und geben wiederum den Dateinamen ein. Nun sehen Sie alle Datenfelder zusammen mit einer Nummer aufgelistet. Um ein bestimmtes Feld zu löschen, geben Sie seine Nummer ein. Achtung: Wenn Sie ein Feld löschen, löschen Sie auch alle Inhalte dieses Feldes. Falls Sie doch nichts löschen wollen, und um das Löschen zu beenden, geben Sie »0« ein. Auch hier müssen Sie »Geänderte Datei sichern als« verwenden, um die neue Datei zu sichern.

Etwas komplizierter wird es, wenn Sie die Reihenfolge der Felder in einer Datei ändern wollen. Dazu müssen Sie zuerst »Anzahl Datenfelder erhöhen« wählen und den Namen IHrer Datei eingeben. Auf die nun folgende Frage geben Sie »0« ein und drükken <RETURN>. Nun wählen Sie den Menüpunkt »Reihenfolge der Felder ändern«. Das Programm zeigt Ihnen die Felder Ihrer Datei mit Nummern an. Hier lassen sich jeweils zwei Felder der Datei gegeneinander vertauschen. Dazu geben Sie »J« ein und dann, durch Komma getrennt, die Nummern der zu vertauschenden Felder. Ein Beispiel: Verwenden Sie

```
1,5
```

um die Felder 1 und 5 zu vertauschen.

Geben Sie nun wieder »J« ein, können Sie wieder zwei Felder vertauschen und so weiter. Mit »N« auf die Frage »Willst Du zwei Felder vertauschen« bricht die Funktion ab; speichern Sie Ihre Datei mit »Geänderte Datei sichern als«.

Wollen Sie die Feldanzahl einer Datei erhöhen oder reduzieren, müssen Sie »Programm neu starten« wählen und diese Operation wie beschrieben ausführen.

Leider gibt es bei einem 512-KByte-Amiga manchmal Speicherprobleme mit »UniUtil«. Es hat sich am besten bewährt, keine anderen Programme gleichzeitig laufen zu lassen und das Programm von der Workbench aus zu starten.

Bereits nach kurzer Einarbeitungszeit werden Sie feststellen, daß Uni-Datei Ihnen die Arbeit sehr einfach macht. Die einzelnen Funktionen sind sauber durchdacht und so schnell, daß Sie nur mit geringen Wartezeiten rechnen müssen. Dazu kommt noch der Vorteil, daß Sie – dank Basic – vollen Zugriff auf alle Funktionen haben und diese nach Ihren Wünschen anpassen können. Also: Legen Sie los. *kn*

```
Wie breit darf jedes Datenfeld gedruckt werden ?
                                (0 = nicht drucken)
Herr/Frau      :  20
Vorname        :  20   Sie haben eine Zeilenbreite
Name           :  20   von 360 Zeichen eingestellt.
Strasse        :  20
PLZ            :  20
Ort            :  20
Bezirk         :  20                    [↑]
Vorwahl        :  20
Telefon        :  20
Bemerkung 1    :  20         [±] [<] [O.K.] [>]
Bemerkung 2    :  20
Kundennummer   :  20                    [↓]
Auftragsnr.    :  20
Let. Auftrag   :  20
Kredit         :  20
Rabatt         :  20
Skonto         :  20
Satz OK        :  20

Anzahl Spalten zwischen      1          [<] [±] [>]
Datenfelder (in Zeichen)
```

Bild 5. Dieses Fenster verwenden Sie, um aus Ihren Daten Tabellen zu machen.

```
*** Uni-Datei Druckerparameter ***

Linker Rand       (in Zeichen)      5   [<] ± [>]
Etikettenhöhe     (in Zeilen)      18   [<] ± [>]
Anzahl Zeilen zwischen Etiketten    1   [<] ± [>]
Informations-Zeile drucken              [Nein]
Zeilenabstand in Zoll                   [1/6]
Druckqualität (NLQ/Draft)               [Draft]
Schrift (Pica/Elite/schmal)             [Pica]
Feldnamen mitdrucken                    [Ja]
Datenfelder nebeneinander               [Nein]
  in eine Zeile drucken
                                        [O.K.]
```

Bild 4. Hier werden die Druckerparameter eingestellt

Checksummer

Dieses Programm ist unentbehrlich beim Abtippen unserer Listings. Es hilft, Tippfehler zu vermeiden und spart viel Zeit.

Ein längeres Listing ohne Fehler abzutippen ist (fast) unmöglich. Aus diesem Grund haben wir in Ausgabe 3/88 des AMIGA-Magazins eine Eingabehilfe – den Checksummer »Checkie 42« veröffentlicht. Die hier vorgestellte Version 1.1 enthält erweiterte Funktionen und bietet mehr Komfort. Damit möglichst viele unserer Leser dieses Programm auch tatsächlich anwenden, haben wir es möglichst kurz gehalten und in einer Sprache programmiert, die alle Abtipper besitzen: Amiga-Basic.

Die Form der Listings

Die Listingzeilen bestehen aus einer bis zu vierstelligen Zeilennummer, der zwei- bzw. dreistelligen Prüfsumme und der eigentlichen Programmzeile. Beispiel:

```
10 TTO print "Hallo!"
|   └── Prüfcode
└────── Zeilennummer
```

Nach einer Leerstelle im Anschluß an die Zeilennummer stehen bis zu drei Zeichen Prüfcode. Die einzelnen Zeichen können sein eine Ziffer (»0« bis »9«), ein kleiner Buchstabe (»a« bis »z«) oder ein Großbuchstabe (»A« bis »Z«). Die ersten beiden Zeichen der Prüfsumme sind der eigentliche Prüfcode. Im dritten Zeichen ist die Spaltenposition der ersten »Nicht-Leerstelle« verschlüsselt. Das ist für diejenigen Anwender interessant, welche die Struktur des Listings, also die Einrückungen durch Leerzeichen, übernehmen wollen. Ist dies nicht Ihre Absicht, können Sie die Eingabe der Checksumme schon nach den ersten beiden Zeichen mit <Return> abschließen. Bei Checkie 42 muß die Groß- und Kleinschreibung so wie im Listing abgedruckt übernommen werden.

Der Umgang mit Checkie 42

Nach dem Start fragt das Programm nach einem Dateinamen. Unter dem angegebenen Namen speichert Checkie 42 die eingegebenen Listingzeilen ab. Existiert bereits eine Datei mit diesem Namen auf der Diskette, so haben Sie mit der Abfrage »Nur Checksummer ausgeben?« zwei Möglichkeiten:

<j> Ausgabe der Datei mit Checksumme auf den Bildschirm oder den Drucker.

<n> Einlesen der Programmzeilen aus der vorhandenen Datei und Eingabe der Checksumme mit der Tastatur.

> **Eingabehinweis**
>
> Geben Sie »Checkie 42« (Version 1.1) bitte mit Checkie 42 aus der AMIGA-Ausgabe 3/88 oder 12/87 ein. Sollten Sie die alte Eingabehilfe nicht besitzen, so tippen Sie das Listing für die Version 1.1 im normalen Basic-Editor ohne Prüfsummen und Zeilennummern ab.

Beide Alternativen sind gedacht für Anwender, die ein Listing nicht mit dem Zeileneditor des Checkie, sondern mit einem schnelleren und/oder komfortableren Editor ihrer Wahl – z. B. dem Editor von Amiga-Basic (mit »..",a« speichern) erfaßt haben.

Checkie 42 errechnet nach der Eingabe <j> die Prüfsummen Ihres Textes und Sie können diese dann mit dem Listing im AMIGA-Magazin vergleichen. Bei der Ausgabe auf den Bildschirm schreibt das Programm die Programmzeilen inklusive Checksummen zusätzlich in eine Datei auf Diskette mit dem Zusatz ».chk«. Diese können Sie später z. B. mit dem CLI-Befehl TYPE ohne erneute Berechnung der Prüfsumme noch einmal ausgeben.

Haben Sie »Nur Checksumme ausgegeben?« mit »n« beantwortet, dann können Sie dem Programm den Vergleich überlassen, in dem die Frage »Eingabe aus Datei« mit »j« beantwortet wird. Dann brauchen Sie nur noch die Checksummen eingeben. Der Checksummer holt sich die Zeile aus der angegebenen Datei statt von der Tastatur. Entspricht die eingegebene Prüfzahl nicht der errechneten, kann die Zeile gleich korrigiert werden.

Beantworten Sie obige Frage mit »n«, zählt Checkie die in der Datei vorhandenen Zeilen und wartet mit der Zeilennummer »Anzahl+1« auf die Eingabe einer neuen Zeile. Alle weiteren Eingaben hängt das Programm an die bestehende Datei an. Diese Funktion ist sinnvoll, wenn Sie ein Listing in mehreren Teilen abtippen wollen.

So tippen Sie Listings ab

Haben Sie anfangs einen Dateinamen eingegeben oder Sie hängen neue Zeilen an eine bestehende Datei an, dann arbeiten Sie im normalen Eingabemodus. Checkie 42 schlägt dabei eine Zeilennummer vor und wartet auf die Prüfsumme. Nach Eingabe derselben taucht der Cursor zwischen den zwei Trennstrichen auf. Dort muß nun die Zeile »ohne« Zeilennummer und Prüfsumme eingegeben werden. Nach Betätigen der Taste <Return> berechnet Checkie die Prüfsumme. Leerstellen vor und hinter der Programmanweisung werden ignoriert. Stimmen Programmzeile und Prüfsumme mit derjenigen im Listing überein, speichert der Checksummer die Eingabe ab und wartet auf die nächste Zeile.

Einfügemodus

Wahrscheinlich wird eine abgetippte Zeile mal einen Fehler enthalten. Checkie 42 positioniert den Cursor dann an den Anfang der Zeile und wartet auf die korrekte Eingabe. Korrekturen lassen sich mit der Backspace- oder Delete-Taste durchführen. Um Zeichenfolgen einzufügen, kann kurzfristig mit <F2> der Einfügemodus eingeschaltet werden. Dieser Modus sollte allerdings nach der Fehlerkorrektur wieder ausgeschaltet werden, da er die Eingabe verlangsamt.

Sonderfall: Prüfsumme ignorieren

Möchten Sie z. B. eine Kommentarzeile nicht »original« übernehmen, läßt sich trotz einer falschen Prüfsumme eine Übernahme der Zeile mit der Funktionstaste <F6> erzwingen. Sie können damit aber auch falsche Programmzeilen übernehmen. Verwenden Sie deshalb die Taste <F6> nicht gewohnheitsmäßig. Der Checksummer teilt Ihnen nach Beenden des Programms mit, wieviel Zeilen er ungeprüft übernommen hat.

Prüfsumme und Zeilennummer ändern

Natürlich kann es auch vorkommen, daß die Programmzeile zwar richtig abgetippt wurde, sich bei der Prüfsumme aber ein Fehler eingeschlichen hat. Nach Betätigen von <F1> kann die Prüfsumme korrigiert werden. Während der Eingabe der Prüfsumme läßt sich mit <F7> die vom Programm vorgeschlagene Zeilennummer verändern. Damit können Sie gezielt nur bestimmte Teile eines Listings übernehmen.

Haben Sie eine mit einem anderen Editor geschriebene Programmdatei überprüft und nur in wenigen Zeilen Fehler festgestellt, lassen sich durch Vorgabe der Nummern diese Zeilen gezielt ändern. Bei Angabe der Zeilennummer in aufsteigender Reihenfolge benötigt das Programm übrigens erheblich weniger Zeit für die Suche der Zeilen in der jeweiligen Datei. Um die versehentliche Übernahme fehlerhafter Zeilen zu verhindern, sperrt das Programm bei fehlender Übereinstimmung der Prüfsummen die Taste <F7> (Änderung der Zeilennummer).

Fehlerfrei abtippen

Eingabe beenden

Die Kombination <Ctrl-e> beendet den Programmlauf nach vollständiger Eingabe des Listings oder für eine Unterbrechung.

Am Schluß noch ein Tip für alle Leser, denen unser Basic-Editor zu langsam ist. Die Berechnung der Prüfsummen erfolgt im Unterprogramm »CalcSumme«. Dieser Teil ist sehr einfach in schnellere Sprachen, wie beispielsweise C, umsetzbar.

Wer schon einmal Fehler in einem abgetippten Listing gesucht hat, der weiß, wie frustrierend diese Arbeit sein kann. Nutzen Sie deshalb den »Checkie 42«. Sie sparen viel Zeit und müssen sich nicht dauernd auf die Suche nach tückischen Fehlern begeben.

Dieter Behlich/kn

```
CqO REM ***********************************
KF  REM *** Checksummer: Checkie 42       ***
4c  REM ***         Version 1.1           ***
Ft  REM ***********************************
Z7  Start:
W42   GOSUB Init
Pn    GOSUB OpenDatei
mb    IF dn$="" THEN Ende
b2    GOSUB Bild
OwO NeueZeile:
K42   GOSUB loeschen
zT    GOSUB EingabeSumme
pt    IF FEnde=wahr THEN Ende
MVO Wiederholung:
sQ2   GOSUB EingabeZeile
sw    IF FEnde=wahr THEN Ende
mO    GOSUB CalcSumme
fB    IF FZok = falsch THEN GOTO Wiederholung
oC    GOSUB Uebernahme
iv    GOTO NeueZeile
mAO Ende:
7S2   GOSUB fertig
72    END
UGO Init:
fD2   wahr=-1
fy    falsch=0
Kh    LZeile=240  : REM Anzahl Zeichen/Zeile
tz    LBZeile=60  : REM Anzahl Zeichen/Bildschirmzeile
OG    AnzBZeilen=LZeile/LBZeile
N1    zx=6 : zy=14   : REM Position Zeile
PC    sx=20 : sy=5   : REM Position Checksumme
mF    AnzCsZ=3       : REM Anzahl Ziffern/Checksumme
QF    DIM z(LZeile)
05    DIM cs(AnzCsZ*2)
iz    cs(AnzCsZ)=0   : REM Zeilenstart
gv    a=0 : b=0 : c=0  : REM Hilfsvariablen
AI    i=0 : j=0 : k=0  : REM Zählvariablen
XL    FZok = wahr : FCz3=100
Wn    Checkfile=0  : Zeile=1
Bb    READ Faktor(i)
X1    WHILE Faktor(i)<>0
ga4     i=i+1
cm      READ Faktor(i)  : REM Faktorenreihe
9x2   WEND
Gm    AnzFak=i
RB    DATA 2,3,4,5,6,0
7j    RETURN
PWO OpenDatei:
BR2   ON ERROR GOTO Dateifehler
MS    CLS
Vu    LOCATE 3,5:dn$="":INPUT "Dateiname: ";dn$
Fp    OPEN dn$ FOR INPUT AS #1
KU    IF FFehler=53 THEN NeueDatei
4s    LOCATE 5,5:PRINT "Nur Checksummen ausgeben? (j/n) ":e$=INP
      UT$(1)
gM    IF e$="j" THEN
Kq4     LOCATE 7,5:e$="":PRINT "Bildschirm oder Drucker? (b/d) "
        :e$=INPUT$(1)
Ku      IF e$="d" THEN
KM7        OPEN "prt:" FOR OUTPUT AS #4
YH4     ELSE
7Y6       OPEN dn$+".chk" FOR OUTPUT AS #4
vo4     END IF
RR      WHILE NOT EOF(1)
yJ6       LINE INPUT#1,e$
fg        apos=LEN(e$)
to        z(apos)=32
4L        FOR i=1 TO apos
sq8         z(i-1)=ASC(MID$(e$,i,1))
Xn6       NEXT i
cE        GOSUB CalcSumme
vp        i=0
Rh        WHILE z(i)=32 : i=i+1 : WEND
4v        IF FCz3=i THEN
f98         cs(6)=-16    : REM Leerstelle
nW6       ELSE
XA8         cs(6)=i : FCz3=i
A36       END IF
2o        PRINT USING "####";Zeile; : PRINT " ";
Wf        PRINT#4,USING "####";Zeile; : PRINT#4," ";
mC        FOR i=4 TO 6
7d8         IF cs(i)>35 THEN cs(i)=cs(i)+6
nS          IF cs(i)>9 THEN cs(i)=cs(i)+7
jy          PRINT CHR$(cs(i)+48);
8L          PRINT#4,CHR$(cs(i)+48);
n36       NEXT i
19        PRINT " ";e$
tl        PRINT#4," ";e$
Qm        Zeile=Zeile+1
rf4     WEND
IF      CLOSE 1 : CLOSE 4
H4      dn$=""
Fv      PRINT:PRINT "Taste drÜcken"
5r      e$=INPUT$(1)
rT      RETURN
SL2   END IF
70    LOCATE 7,5:PRINT"Eingabe aus Datei? (j/n) ":e$=INPUT$(1)
L1    IF e$="j" THEN
u24     Checkfile=1
08      CLOSE 1
eX      GOSUB backup
WZ      OPEN dn$+".bak" FOR INPUT AS #1
Sk      OPEN dn$ FOR OUTPUT AS #2
Oc      RETURN
bU2   END IF
77    WHILE NOT EOF(1)
eP4     LINE INPUT#1,e$
j5      Zeile=Zeile+1
Ay2   WEND
YI    CLOSE 1
8AO NeueDatei:
aM2   OPEN dn$ FOR APPEND AS #2
91    RETURN
7QO Dateifehler:
Og2   FFehler=ERR
PB    IF FFehler=53 THEN RESUME NEXT
vy    ON ERROR GOTO 0
aDO backup:
yY2   OPEN dn$ FOR INPUT AS#1
Sj    OPEN dn$+".bak" FOR OUTPUT AS #2
8F    flen=LOF(1)
ih    WHILE flen>=32000
i14     e$=INPUT$(32000,1)
tL      PRINT#2,e$;
QE2   WEND
e7    IF flen THEN
SB4     e$=INPUT$(flen,1)
xP      PRINT#2,e$;
zs2   END IF
zg    CLOSE 2  : CLOSE 1
R3    RETURN
dOO Bild:
f12   CLS
Gi    LOCATE 2,10
yY    PRINT ">>>>>>> Checkie 42 - Der ultimative Checksum
      mer! <<<<<<<"
xR    LOCATE 13,zx
Le    FOR i=1 TO 2
kx4     FOR j=1 TO LBZeile
HR6       PRINT "-";
hy4     NEXT j
H1      LOCATE 13+AnzBZeilen+1,zx
hx2   NEXT i
Hi    LOCATE 5,30:PRINT "<CTRL e> = Programm beenden"
vd    LOCATE 7,8:PRINT "<F2> = EinfÜgemodus ein"
uQ    LOCATE 7,40:PRINT "<F1> = Checksumme ändern"
1H    LOCATE 9,40:PRINT "<F6> = Zeile speichern"
ME    LOCATE 9,8:PRINT "<F3> = Zeile löschen"
v5    LOCATE 11,8:PRINT "<F7> = Zeilennummer ändern"
jL    RETURN
5fO loeschen:
w82   LOCATE zy,1
8S    FOR i=1 TO AnzBZeilen
jf4     PRINT TAB(zx);SPACE$(LBZeile)
t92   NEXT i
OO    FOR i=0 TO LZeile
V84     z(i)=32
wC2   NEXT i
3F    LOCATE zy,1
om    PRINT USING "####";Zeile
dy    apos=0
vX    RETURN
v5O EingabeSumme:
va2   Farbe=0 : i=1
AB    LOCATE sy,sx+1
Xm    PRINT "..."
O4O blinken:
aC2   Farbe=Farbe XOR 1
```



```
QH    COLOR Farbe,0
P6    LOCATE sy,sx-15
1n    PRINT "Checksumme:"
ni    LOCATE sy,sx+i
ya    e$=INKEY$
lT    IF e$="" THEN blinken
Xp    COLOR 1,0
dl    e=ASC(e$)
fM    IF e=5 THEN
KU4     FEnde=wahr   : REM <CTRL e>
hf      i=AnzCsZ+1   : REM fertig
Cc2   ELSEIF e=135 THEN
dc4     GOSUB NeuZeile
Bz      e=135
LB2   ELSEIF e=8 THEN
4h4     IF i>1 THEN
sZ6       i=i-1      : REM <BS>
XY        LOCATE sy,sx+i : PRINT "."
un4     END IF
Tn2   ELSEIF e=13 THEN
KZ4     IF i=AnzCsZ THEN i=AnzCsZ+1 : REM <CR>
cL2   ELSE
ra4     IF e>47 AND e<58 THEN
vi6       e=e-48                    : REM 0-9
AI4     ELSEIF e>64 AND e<91 THEN
Rv6       e=e-55                    : REM A-Z
jX4     ELSEIF e>96 AND e<123 THEN
Zx6       e=e-61                    : REM a-z
jS4     ELSE
yz6       GOTO blinken              : REM weder noch
6z4     END IF
JY      PRINT e$;
ve      cs(i)=e
D7      i=i+1
A32   END IF
wn    IF i<=AnzCsZ THEN blinken
f50 ESEnde:
2K2   COLOR 1,0
Oh    LOCATE sy,sx-15
cO    PRINT "Checksumme:"
gI    RETURN
aE0 NeuZeile:
dU2   IF FZok = wahr THEN
P44     NeuZeile=0
nn      WHILE e<>13 OR NeuZeile=0
Ux6       LOCATE zy,1:PRINT USING "####";NeuZeile;
oF        e=ASC(INPUT$(1))
gP        IF e>47 AND e<58 THEN NeuZeile=NeuZeile*10+e-48
Jn        IF NeuZeile > 9999 THEN e=8
in        IF e=8 THEN NeuZeile=INT(NeuZeile/10)
vj4     WEND
kh      IF Checkfile THEN
tQ6       IF NeuZeile < Zeile THEN
yy8         WHILE NOT EOF(1)
VGA           LINE INPUT#1,e$
cd            PRINT#2,e$
1p8         WEND
MH          CLOSE 1 : CLOSE 2
fY          GOSUB backup
Xa          OPEN dn$+".bak" FOR INPUT AS #1
T1          OPEN dn$ FOR OUTPUT AS #2
9t          Zeile=1
cV6       END IF
Yp        WHILE (NeuZeile > Zeile) AND (NOT EOF(1))
fQ8         LINE INPUT#1,e$
mn          PRINT#2,e$
17          Zeile=Zeile+1
CO6       WEND
cR        IF EOF(1) THEN
bL8         CLOSE 1
81          NeuZeile=Zeile
uN          LOCATE zy,1:PRINT USING "####";NeuZeile;
9G          Checkfile=0
ng6       END IF
ch4     END IF
vV      Zeile=NeuZeile
qJ2   END IF
Ht0 RETURN
1H  EingabeZeile:
n72   x=cs(AnzCsZ)
BW0 weiter:
ga2   cy=zy+INT(x/LBZeile):cx=zx+(x MOD LBZeile)
vy    LOCATE cy,cx
o6    COLOR 0,1
mI    PRINT CHR$(z(x));
y1    LOCATE cy,cx
68    IF x>apos THEN apos=x
xx    IF Checkfile AND FZok THEN
w14     IF EOF(1) THEN
OR6       Checkfile=0 : CLOSE 1
jS4     ELSE
ix6       e$=INPUT$(1,1)
6z4     END IF
mV2   ELSE
Q24     e$=INKEY$
922   END IF
gp4     IF e$="" THEN weiter
0I      COLOR 1,0
OW      PRINT CHR$(z(x));
CF      LOCATE cy,cx
8G      e=ASC(e$)
UR2   IF ((e AND 127)<32) OR e=127 THEN Controlcode
op    IF imode THEN GOSUB insert
I6    PRINT e$
Ki    z(x)=e : e=30
3D0 Controlcode:
gF2   IF e=13 OR e=10 THEN
1N4     RETURN
092   ELSEIF e=30 THEN
2p4     a=1
Ea2   ELSEIF e=29 THEN
034     a=LBZeile
6G2   ELSEIF e=31 THEN
DD4     a=-1
Gb2   ELSEIF e=28 THEN
Rg4     a=-LBZeile
9s2   ELSE
UX4     GOTO noCrs
WP2   END IF
hz    x=x+a
5Z    IF x>=0 AND x<LZeile THEN weiter
tD    x=x-a
HB    GOTO weiter
kE0 noCrs:
dN2   IF e=8 THEN
1s4     IF x>0 THEN
Kk6       x=x-1
AM        LOCATE zy+INT(x/LBZeile),zx+(x MOD LBZeile)
Bb        FOR i=x TO apos
788         z(i)=z(i+1)
Ja          PRINT CHR$(z(i));
Z0          IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(zx);
FV6       NEXT i
pr        z(apos)=32 : PRINT " "
yR        apos=apos-1
ng4     END IF
H12   ELSEIF e=127 THEN
Kk4     FOR i=x TO apos
GH6       z(i)=z(i+1)
Sj        PRINT CHR$(z(i));
i9        IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(zx);
Oe4     NEXT i
y0      z(apos)=32 : PRINT "
7a      apos=apos-1
Vy2   ELSEIF e=129 THEN
oI4     GOSUB EingabeSumme
sC      x=cs(AnzCsZ)
9U2   ELSEIF e=130 THEN
513    imode=imode XOR 1
tU5      LOCATE 7,28
yZ3    IF imode THEN
1T5      PRINT "aus"
jS3    ELSE
YV5      PRINT "ein"
6z3    END IF
Kg2   ELSEIF e=131 THEN
L54     GOSUB loeschen
```

Listing. Der verbesserte »Checkie 42«. Bitte mit der ersten Version von »Checkie 42« oder ohne Prüfsummen und Zeilennummern eingeben.

(Fortsetzung) ▶

Vorschau

AMIGA-SONDERHEFT 10

AmigaBasic ist eine leistungsfähige Programmiersprache, wenn man sie zu nutzen weiß. Wir bieten Ihnen einen leichten Einstieg in die Welt des Programmierens und zeigen, wie Profis in Basic das letzte Byte aus dem Amiga-Betriebssystem hervorzaubern.

Im Sonderheft 10 erfahren Sie alles über das Einbinden und Ansprechen von System-Routinen und Devices. So finden Sie neben den grundlegenden Kursen auch ausführliche Artikel zu VSprites, Hardware-Sprites, Bobs, Superbitmaps, Double Buffering, Dualplayfield, Copper, Gameport Device und vieles mehr.

Geben Sie Ihren Programmen eine professionelle Note mit einem File-Requester, der auch höchsten Ansprüchen genügt. Sie finden ihn und andere hilfreiche Sub-Programme zum einfachen und problemlosen Einbinden in Ihre Projekte in der nächsten Ausgabe.

Wollen Sie Ihren Basic-Programmen auf die Sprünge helfen? Wir stellen die drei verbreitetsten Compiler bzw. Basic-Dialekte – HisoftBasic, AC Basic und GFA Basic – vor und zeigen, wo ihre Stärken und Schwächen liegen.



Das zehnte Sonderheft liegt ab 28.2. an Ihrem Kiosk.

```
3N       x=cs(AnzCsZ)
7Y2    ELSEIF e=134  THEN
bD4      RETURN
PR2    ELSEIF e=5 THEN
ZT4      FEnde=wahr
eG       RETURN
F82    END IF
xr     GOTO weiter
If0  insert:
oJ2    IF apos>x THEN
3o4      FOR i=apos TO x STEP -1
Qt6        z(i+1)=z(i)
q64      NEXT i
PH       z(x)=32
Ny       apos=apos+1
Kp       IF apos=LZeile THEN apos=apos-1:z(LZeile)=32
uK       FOR i=x TO apos
1I6        PRINT CHR$(z(i));
Hi         IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(zx);
xD4      NEXT i
yA       LOCATE zy+INT(x/LBZeile),zx+(x MOD LBZeile)
UN2    END IF
vX     RETURN
cX0  CalcSumme:
3s2    a=0 : b=0 : c=0
z2     IF e=134 THEN
pv4      FZok=wahr
WX       FF6=FF6+1
Gz2    ELSE
hv4      WHILE z(apos)=32 AND apos>0
pI6        apos=apos-1
9x4      WEND
Ni       IF apos>0 THEN
036        WHILE z(c)=32
Bt8          c=c+1
D16        WEND
jc4      END IF
EJ       FOR i=c TO apos
Y26        j=(i-c) MOD AnzFak
Qs         k=(i+1-c) MOD AnzFak
s1         a=a+((z(i) AND 127)-32)*Faktor(j)
2E         b=b+((z(i) AND 127)-32)*Faktor(k)
Ka4      NEXT i
pA       cs(4)=a+Zeile-INT((a+Zeile)/62)*62
4N       cs(5)=b+Zeile-INT((b+Zeile)/62)*62
4Q       FZok=(cs(1)=cs(4)) AND (cs(2)=cs(5))
tm2    END IF
Kw     RETURN
pM0  Uebernahme:
xD2    FOR i=0 TO apos
ih4      PRINT#2,CHR$(z(i));
TJ2    NEXT i
VZ4      PRINT#2,""
5R2    Zeile=Zeile+1
R3     RETURN
N90  fertig:
MJ2    IF Checkfile THEN
ZZ4      WHILE NOT EOF(1)
6r6        LINE INPUT#1,e$
DE         PRINT#2,e$
cQ4      WEND
Ok       CLOSE 1
922    END IF
5q     CLOSE 2
nt     CLS
Rg     LOCATE 12,35
WL     PRINT "F E R T I G !!!"
I9     LOCATE 20,1
D1     IF FF6<>0 THEN
Ci4      PRINT "ACHTUNG!!!  ";
9H       PRINT FF6;" Zeile(n) wurde(n) ungeprÜft gespeichert."
IB2    END IF
jL     RETURN
(C) 1988 M&T
```

Listing. Der verbesserte »Checkie 42« (Schluß)

Stoff für Ihren Amiga



HiSoft-Basic-Compiler

Mit HiSoft Basic gibt es endlich auch einen Basic-Compiler für den Amiga. Der interaktive Editier-, Kompilier- und Laufzeitzyklus entspricht dem eines Interpreters.

Der integrierte Editor erlaubt eine komfortable Eingabe. HiSoft Basic unterstützt die Eigenschaften des Amiga mit Fenstern, Grafik-Kommandos, Sprite-Handling und Maschinenzugriffen während des Gebrauchs von Bibliotheken. Es ist voll kompatibel mit Amiga-Basic, andere Standarddialekte für den PC, wie z.B. Microsoft QuickBasic, und den Atari ST können mit geringen Modifikationen kompiliert werden. Rekursive Unterprogramme und Funktionen sind möglich.

Eine Anzahl strukturierter Ausdrücke wie z.B. WHILE...WEND, DO...LOOP UNTIL und SELECT...CASE lassen Sie jede Programmierhürde meistern. Die Größe von Variablen ist nicht beschränkt. Aussagekräftige Fehlermeldungen und Korrekturmöglichkeiten tragen zur komfortablen Handhabung bei.

Hardware-Anforderungen:
Amiga 500, 1000 oder 2000 mit Kickstart 1.2 oder höher.
3½"-Diskette,
Bestell-Nr. 54127
DM 179,–*
(sFr 161,– /öS 1790,–*)

Devpac Assembler 2.0

Devpac 2.0 ist ein Entwicklungspaket für den Amiga mit komfortablem Editor/Assembler, symbolischem Debugger und Linker zum Einbinden von Hochsprache-Modulen.

GenAm ist ein 68000er-Makroassembler mit integriertem Bildschirmeditor, der bis zu 75.000 Zeilen pro Minute assemblieren kann. Der 2-Paß-Assembler erzeugt sowohl linkbaren als auch direkt ausführbaren Code. Er unterstützt lokale Labels, die Signifikanz beträgt bis zu 127 Zeichen. Makros können bis zu 36 Parameter beinhalten und – Rekursion inbegriffen – so tief verschachtelt werden, wie Speicherplatz vorhanden ist. MonAm, der Debugger, erlaubt das Setzen von Breakpoints, das Disassemblieren auf Diskette und noch weitere Features, die das Debuggen zum Vergnügen machen.

Hardware-Anforderungen:
Amiga 500, 1000 oder 2000 mit mindestens 512 Kbyte, ein Diskettenlaufwerk.
3½"-Diskette,
Bestell-Nr. 54131
DM 149,–*
(sFr 135,–*/öS 1490,–*)

Devpac-Assembler 2.0 im Test! Amiga-Magazin 4/89: Eines der besten Programme seiner Art für den Amiga.

SuperED C

Multitasking-fähiger Editor als Programmierumgebung für den Aztec-C-Compiler (V 3.6).
Bestell-Nr. 54139
DM 39,–*
(sFr 35,–*/öS 390,–*)

* Unverbindliche Preisempfehlung

INFO-COUPON

Bitte senden Sie mir weiteres Informationsmaterial zu
❑ Devpac-Assembler ❑ HiSoft-Basic-Compiler

2211/907

Name ______________________

Straße ______________________



PLZ/Ort ______________________

Bitte ausschneiden und einsenden an: Markt&Technik Verlag AG, Buch- und Software-Verlag, Hans-Pinsel-Straße 2, 8013 Haar bei München

Ami SH 9

Markt&Technik
Zeitschriften · Bücher
Software · Schulung